

**Line Processor: A Device for Amplification of Display Terminal
Capabilities for Text Manipulation**

This is a paper that was submitted to the 1974 NCC (Conference to be held May 1974).

Line Processor: A Device for Amplification of Display Terminal
Capabilities for Text Manipulation

**Line Processor: A Device for Amplification of Display Terminal
Capabilities for Text Manipulation**

**LINE PROCESSOR: A DEVICE FOR AMPLIFICATION OF DISPLAY TERMINAL
CAPABILITIES FOR TEXT MANIPULATION**

By:

Donald I. Andrews

Augmentation Research Center

Stanford Research Institute

333 Ravenswood

Menlo Park, California

(415) 326-6200 ext. 3268, 3630

2

3

3a

3b

3c

3d

3e

3f

PROGRAM AREA: Computer Architecture and Hardware

4

Donald I. Andrews

ABSTRACT:

5

The Augmentation Research Center (ARC) of Stanford Research Institute has developed a microcomputer-based device that makes any of a class of alphanumeric display terminals useful as a high quality two-dimensional interactive workstation. These workstations serve ARC's interactive information manipulation system, NLS (online system).

5a

The device supports a mouse pointing device and a five-finger keyset, and requires no hardware modifications either to the display terminal or to the main computer.

5b

This paper describes the NLS workstations, the Line Processor design and operation, terminal requirements, and other possible applications of the Line Processor concept.

5c

KWIK INDEX WORDS:

6

Augmentation, microcomputer, mouse, keyset, NLS

6a

7

Donald I. Andrews

INTRODUCTION

7a

The Augmentation Research Center (ARC) at Stanford Research Institute has been developing an interactive computer system to enhance the effectiveness of people doing intellectual work. The development of the system, called NLS, has been going on for a number of years.

7a1

NLS is a highly interactive system that joins many capabilities. It supports structured text files, very flexible editing, techniques for viewing and studying, arbitrary word processing and document production, and aids to applications and system programming. (For a general description of ARC's goals and efforts see reference 1; it contains an extensive reference list to related papers.)

7a2

The system can be operated from a range of terminal types: from typewriter terminals to high speed graphics displays. Our main effort has been to develop a carefully human engineered display oriented system. The display version of the system introduces the user to the world of two-dimensional computer interaction, which is much more natural than the one-dimensional mode forced upon typewriter device users [2].

7a3

The NLS program runs as a subsystem of a TENEX timesharing system on ARC's Digital Equipment Corporation PDP-10. The PDP-10 is connected to the ARPA network and many of our users gain access via that network [3,4].

7a4

We will describe the NLS workstation and then explain how

Donald I. Andrews

we were able to make workstations from inexpensive alphanumeric video display terminals at a relatively low cost by developing the Line Processor device. The NLS workstation and associated problems will demonstrate the usefulness of the Line Processor concept.

7a5

NLS WORKSTATION ENVIRONMENT

7b

NLS users employ two special input devices continuously: a mouse and five-finger keyset. We use the term "workstation" to mean a display with standard keyboard and these devices, arranged on a special table in a convenient manner for effective working. (See Figure 1.)

7b1

The two unusual workstation input devices, the mouse and keyset, are commercially available separately, but are offered as standard options on only a few products. (Mice and keysets can be purchased from Imlac, Cybernex and Computer Displays.)

7b2

The five-finger keyset has five long keys as shown in Figure 2. The user rests his fingers lightly on the keys and strikes chords to input characters. The typical user learns enough binary codes in a couple of hours to do useful work.

7b3

The user can point to characters on the screen by rolling the mouse on any flat surface. Potentiometers are connected to the orthogonal wheels on the mouse, and they provide analogue signals that are translated into digital X and Y position coordinates by a two channel A to D convertor. (See Figure 3.)

7b4

The user can operate three buttons on the mouse to confirm

Donald I. Andrews

commands, abort commands, provide a case shift for the keyset and perform other programmable functions. The workstation has no other special function keys or buttons.

7b5

With the right hand on the mouse, and the left hand on the keyset, the user can edit, change his view and perform other control operations rapidly and easily. He does not remove his hands from the mouse and keyset, except to type large amounts of text. The mouse and keyset are further described in references [5] and [6].

7b6

GOALS AND PROBLEMS

7c

For several years, we have been wanting to make NLS available to remote users on inexpensive displays. The primary problem has been commercial availability of adequate displays. At ARC, we have a custom-built display system that meets our needs, but is not available to others [7].

7c1

Very few displays on the market meet our requirements, and it is generally very expensive or impossible to interface devices -- such as a mouse and keyset -- to existing products. At this time, the only commercially available product with mouse and keyset that meets our requirements is a minicomputer-based display terminal, the IMLAC PDS-1. It makes a satisfactory NLS workstation, but appropriately configured it costs about \$18,000. We would like to see workstations available for about \$5,000.

7c2

Many low-cost, alphanumeric video display terminals on the

Donald I. Andrews

market today are attractive and almost meet our needs. However, they are generally designed as replacements for typewriter terminals, IBM 2265's, or other existing terminals. These video display terminals are fine for their intended purpose, but with respect to our needs, they have the following shortcomings:

7c3

They do not have provisions for adequate graphical pointing devices such as a mouse. They generally have a cursor moved by one of four cursor positioning keys, but this type of operation is too slow and awkward for interactive pointing and selecting.

7c3a

They usually do not meet our general requirements such as: a full ASCII character set (including upper and lower case), adequate number of lines and characters per line, and necessary control functions (such as delete line). We will discuss these requirements later.

7c3b

In addition, we have the problem of producing a software driver to support each type of terminal. There is no industry standard for the terminal function codes, such as delete line. What is worse, there is no general standard as to how some of the functions are carried out on the screen. Display terminal manipulations are relatively complex; Supporting many types of terminals would amount to a considerable software effort and a large software maintenance problem. We have written software

Donald I. Andrews

drivers for many types of terminals, and the software expense has been large.

7c4

SOLUTION

7d

To solve the problems mentioned above, we built a device based on a four-bit microcomputer, which we call a Line Processor. The device resides in the transmission line between the display terminal and the main computer. (See Figure 4.) Since it has processing capabilities, it can appear to the main computer as almost any kind of terminal. Because we knew exactly how we were going to use the terminal, we designed a "virtual alphanumeric workstation," which we call a Line Processor terminal [2].

7d1

The only connections to the display and the main computer are standard RS232 interfaces: There are no special hardware connections either to the display or to the main computer. Remote operation is possible over any full duplex line or computer network. (See Figure 4.)

7d2

The device connects to a keyset and mouse. All the special purpose hardware for these connections is neatly localized. Communication from these special input devices is incorporated in an input protocol that is transmitted over the full duplex communication line.

7d3

The microcomputer can be programmed to work with any alphanumeric display that meets certain requirements. No modifications to the display are necessary.

7d4

Donald I. Andrews

All composite terminals (display terminal plus Line Processor) are logically the same at the point of connection, and as a result, only one software driver is required.

7d5

The microcomputer program does not require any software maintenance expense, since it is treated as a hardware device when the development is completed. On the other hand, the microcomputer program could be changed without altering the Line Processor hardware, if that proved to be useful.

7d6

The reader may wonder why it is more feasible to adapt Line Processor programs to different terminal types than it is to write a software driver for each type of terminal. Briefly, these are the chief considerations:

7d6a

There is no software maintenance associated with the Line Processor.

7d6a1

The microcomputer memory and CPU are cheaper: The microcomputer is better suited to the task.

7d6a2

If other systems are written to make use of Line Processor workstations, they need only one software driver.

7d6a3

REJECTED SOLUTIONS

7e

We considered several alternative solutions to the problems mentioned above. Here are the alternatives that we investigated but did not follow:

7e1

(1) Have a special workstation terminal built by a reputable manufacturer.

7e1a

Donald I. Andrews

We sent out a request to several manufacturers for a proposal to build such a terminal in 1972. There was virtually no interest in serving our needs, and all indications were that special workstation terminals would be very expensive. The manufacturers either were not interested in modifying their standard products or would not address themselves to a relatively small market.

7e1a1

(2) Modify an existing alphanumeric terminal.

7e1b

There are several disadvantages to this approach. Having the modifications made on a production basis and convincing the manufacturer to maintain modified terminals are big obstacles. Also, once the terminal is obsolete or no longer manufactured for any reason, we are faced with our original problems again.

7e1b1

(3) Use a programmable terminal that offers the necessary I/O capabilities.

7e1c

This is exactly what we did by programming the IMLAC PDS-1, but we feel that it results in a workstation that is too expensive. Other programmable terminals that met our general requirements exist in the same price range as the PDS-1; one of our primary objectives was to reduce the price per terminal.

7e1c1

(4) Build a very simple hardware device to interface the mouse and keyset to a communications line.

7e1d

The Line Processor performs several functions that

Donald I. Andrews

need not be performed at the terminal site. For example, mouse tracking, TTY-simulation and display manipulation protocol implementation could be performed in the main computer. This is a feasible idea but the device would not reduce the compute load on the main computer, and it would not reduce the transmission over the terminal-computer connection. Furthermore, it would require more software in the main computer. In a large and fixed system, this may be an advisable approach, but we felt that it was unwise for our configuration.

7e1d1

LINE PROCESSOR FUNCTIONS

7f

There are several important functions that the microcomputer performs to make the combination of display and Line Processor an effective workstation.

7f1

A. Protocol Implementation

7f2

There are several protocols involved. They are identified in Figure 4.

7f2a

The Display Manipulation protocol is exactly the same for every Line Processor workstation and is sent by the applications program to the Line Processor to change the display image. It does not affect the display terminal directly, but is translated by the microcomputer into the Terminal Function protocol.

7f2b

The Display Manipulation protocol is designed to work

Donald I. Andrews

with any alphanumeric terminal with cursor control and line editing functions such as delete line and insert line.

7f2c

The Line Processor talks to the display terminal in the Terminal Function protocol. This is defined by the terminal manufacturer and usually consists of ASCII control codes, or sequences of control codes, interspersed with ASCII text to be written on the display screen.

7f2d

The Line Processor workstation serves as both a timesharing system typewriter terminal and an applications system display output terminal. Hence, there are potentially two streams of output going from the main computer to the Line Processor on the same communication line: the Display Manipulation protocol, and any teletypewriter terminal output that the timesharing system or applications programs send. The teletypewriter output would be generated if the user was using the terminal as a typewriter terminal, or if the user received an error message or some type of system-wide message. These two streams of output are separated by the Line Processor, and TTY-type output is displayed in a TTY-simulation area. This means that the teletypewriter output is not scrambled in with the display output, but it is scrolled -- teletypewriter fashion -- in a small portion of the screen. The applications program has control over the size and

Donald I. Andrews

location of the TTY-simulation area. (The TTY-simulation and window concepts are described in reference [2].)

7f2e

B. Keypad and Mouse Button Transmission (Input Protocol)

7f3

Information from the input devices (keyboard, mouse and keypad) are incorporated into an Input protocol by the Line Processor and sent to the main computer. The protocol is quite simple. Control characters and mouse button changes are sent as short strings of characters that include the mouse tracking spot location (line and character position). Characters from the display's keyboard are sent in unescorted fashion, that is, each single ASCII character is sent.

7f3a

Input from the display terminal's keyboard simply comes into the Line Processor from the full duplex display connection. This consists of ASCII characters that the user has typed on the keyboard. Local editing from the keyboard and batch transmission features within the terminal, if present, are not used for NLS applications.

7f3b

There are three buttons on the mouse. A change of button positions sends in one character to the main computer, which indicates the new three-button status. The applications program knows the state of the mouse buttons and can interpret any subsequent input accordingly. This in effect allows eight different interpretations of keypad and keyboard input [2].

7f3c

Donald I. Andrews

The five-finger keyset input is interpreted as "chords". The normal condition is all-keys-up, and the end of a stroke is indicated by all-keys-up. Any keys that were depressed during the stroke are "OR'ed" together to comprise the chord, which eliminates the problem of striking keys in unison. Thus, a keyset input stroke is a non-zero, five-bit integer. It is converted to an input character in the Line Processor, and sent on to the main computer in unescorted fashion.

7f3d

The five-finger keyset and mouse buttons are gated directly into the microcomputer. These signals are not "clean" and may be "bouncing" for a few milliseconds after a button is pushed. The switches are sampled at a rate at which the bouncing can be detected, and the final reading is not taken until the switches are stable.

7f3e

B. Mouse Tracking

7f4

The Line Processor reads the mouse position from the A to D convertor and "tracks" its movements on the screen with the standard display cursor implemented within the video terminal. This is done by periodically reading the mouse position and, if it has changed, sending a position-cursor command (a part of the Terminal Function protocol) to the display.

7f4a

The tracking allows the user to point to any character on the screen at any time, yet the main computer is only

Donald I. Andrews

informed of the mouse position when the user strikes an appropriate key or button.

7f4b

The mouse moves in a smooth continuous manner, but the tracking spot (cursor) moves from one character position to the next in discrete jumps. The tracking will appear jumpy rather than smooth if the baud rate between the Line Processor and the display is not high enough. Exactly what the baud rate has to be depends on how many characters must be sent for the position cursor function. We have found that 2400 baud is a minimum baud rate with the standard four-character sequence for cursor positioning found on most alphanumeric terminals.

7f4c

D. Display Control

7f5

The Line Processor maintains control of the display because it is the only device sending characters to the display terminal. The applications program in the main computer manipulates the display only by way of the Display Manipulation protocol.

7f5a

The mouse tracking and the TTY-simulation feature mean that the microcomputer must have a fair amount of software logic to keep track of the display's cursor position and know where to move it when necessary.

7f5b

TERMINAL REQUIREMENTS

7g

A display terminal must meet several requirements to form

Donald I. Andrews

an effective workstation with a Line Processor. These requirements are due to either hardware or user considerations:

7g1

1. Hardware Considerations

7g2

It must be possible to perform several kinds of display functions by way of the terminal's RS232 connection. That is, the Terminal Function protocol must be adequate and operable under computer control. We will discuss the actual functions that are required.

7g2a

One of the key functions is positioning the cursor. For the mouse tracking to be satisfactory, it must be a very quick process (within one millisecond), and the cursor must not be displayed in any extraneous positions on the screen while being moved to the destination. The cursor must be addressed by the character and line number of the screen position.

7g2b

The appearance of the cursor must be suitable for tracking. A blinking underline cursor is not very satisfactory since it is not always visible. Some displays implement the cursor by reversing the video in the entire dot matrix of the character in question. The resulting reverse video rectangle that moves around the screen is satisfactory for tracking.

7g2c

It is necessary to have a high speed connection between the Line Processor and the display, as mentioned before. We

Donald I. Andrews

feel that 2400 baud is satisfactory; we prefer 9600 baud because it makes a more effective workstation.

7g2d

The bulk of the display manipulations are done via delete line and insert line functions. These manipulations usually involve positioning the cursor and then issuing the command.

7g2e

Nearly all terminals have a clear screen function, and we require it.

7g2f

We expect carriage return and line feed to work on the display just as they work on a standard teletypewriter device. That is, carriage return moves the cursor to the left margin and line feed moves it to the next line without changing the character position on the line. Although a single "next line" code is often useful, we do not feel it is reasonable for a manufacturer to omit either of these two fundamental functions.

7g2g

When a user selects a character on the screen by pointing to it with the mouse tracking spot and pushing a button, the user program usually responds by "marking" that character for confirmation. The "marking" feature is very important to the interaction process and it has given us a bit of trouble in transforming an alphanumeric display into an effective workstation. We will describe the problem in some detail.

7g2h

The "marking" is done by altering the appearance of

Donald I. Andrews

the character without obliterating it. If the user selected the wrong character he will abort (or "back out of") that selection and select another. The NLS program responds by removing the first mark and putting up another. Hence, it must be possible to "mark" and "unmark" characters on the screen without altering the text in any other way and without re-writing the character. On displays where single characters can be made to blink, reverse in video, or change in some other way, marking will be implemented using that feature. 7g2h1

But, very few displays have the capability of "marking" a character without using up a character position on each side of the character, which is unsatisfactory for our purposes. We have had success by showing the "mark" by "flashing" the cursor at the marked character position at a rate of about three times a second; the cursor is returned to the tracking spot between flashes. This is satisfactory if the display-Line Processor connection is of high enough baud rate (4800 or 9600 baud) and the cursor positioning within the display is fast and does not result in extraneous flashes on the screen. 7g2h2

Clearly, the former type of "marking" is preferred because the latter scheme results in a flashing tracking spot and becomes confusing when there are several "marks"

Donald I. Andrews

on the screen. Probably the most desirable type of "marking" feature would be to make the character blink at 4-5 cps.

7g2h3

Occasionally, it is desirable to write a string on the display with a special appearance, to get the user's attention. We call this "standout mode." Most displays have some kind of standout feature such as blinking, underline, high intensity or reverse video -- we require some kind of standout mode. This differs from marking: in marking an existing character on the screen is altered without rewriting, and in standout mode new text is written on the screen.

7g2i

Of course, all these functions take some finite time for the display to carry out. We would like to see all functions performed as fast as possible. In particular, we expect that all functions except delete line and clear screen will be performed in one millisecond or less. We expect delete line and clear screen to be done in about 5 to 7 milliseconds. We set an upper bound of 120 milliseconds for any function execution time.

7g2j

2. User Considerations

7g3

We consider 24 lines by 64 characters a minimum adequate screen size; however, 27 lines by 80 characters is much more useful. The most desirable would be a full text page of 66 lines by 80 characters. No matter what the screen

Donald I. Andrews

size, we expect the terminal to have enough memory capacity to nearly fill the screen with text. 7g3a

The terminal must display the full ASCII character set, including upper and lower case letters. 7g3b

The keyboard should be a standard typewriter style keyboard with the full ASCII character set, including control characters. Some form of key rollover feature and a comfortable feel are very important. 7g3c

The display output should be readable, easy to look at and flicker free. Our applications are geared toward comfortable day-long use in an office. The terminal should be absolutely quiet and a pleasant thing to work with in all respects. 7g3d

HARDWARE DESCRIPTION 7h

The organization of the Line Processor is outlined in Figure 5. The Line Processor is discussed from a hardware standpoint in reference [8]. 7h1

The heart of the device is an Intel 4004 CPU, which is a four-bit parallel microcomputer in a single MOS integrated circuit chip. The program resides in four Intel 1702A programmable ROM (PROM) chips. Each PROM contains 256 8-bit bytes. These PROM chips are mounted in sockets and can be removed, erased and rewritten in a few minutes. 7h2

The read/write memory consists of 160 four-bit bytes packaged in two Intel 4002 RAM chips. These Intel chips are

Donald I. Andrews

part of Intel's MCS-4 Micro Computer Set, and are easily connected. Both the RAM and PROM chips have I/O ports that are addressed like memory, but they are accessed with different CPU instructions.

7h3

The input devices are multiplexed into one input port. A four-bit address on another port selects the particular four bits of input to be read. The output devices (two serial transmitters) are loaded via two ports and controlled via another port.

7h4

The general configuration is such that Intel 4001 ROM chips can be substituted for the PROM chips, if the production quantity warrants the initial cost of cutting the ROM masks.

7h5

The total cost of all parts is about \$1200 per unit. We intend to find a firm that will manufacture Line Processors and provide nation-wide maintenance. At the writing of this paper, contact has been made with several small manufacturers. It appears that a Line Processor, mouse and keyset should cost under \$2500.

7h6

CONCLUSIONS

7i

The Line Processor approach can effectively upgrade a peripheral device and amplify its capabilities with no actual modifications to the device itself. At the same time, it can optimize the interface between the device and the main computer complex, with respect to both hardware and software.

7i1

In our application of NLS workstation development, the

Donald I. Andrews

concept has proven itself. The alphanumeric terminal needs no modification, and the main computer does not need to know the true nature of the workstation. In other applications, new peripherals could conceivably be interfaced to existing software without modifications to either.

712

A Line Processor coupled with a satisfactory alphanumeric video display terminal results in an NLS workstation that is as effective as a high speed general CRT workstation, except for one respect: the inability to arbitrarily move text on the screen without rewriting it. But, intelligent display manipulation algorithms reduce this problem to the point that users hardly realize that the it exists [2].

713

Other applications for the Line Processor concept are appearing. A microcomputer-based device is being developed at the University of California, Santa Barbara, to make a PLATO IV terminal appear to be a general purpose ASCII terminal for ARPA network use. At ARC, we are considering using a simplified Line Processor to make a line printer with a hostile interface appear to be a serial ASCII device, to avoid purchasing an expensive controller.

714

The microprogram in the Line Processor is still in the developmental stage, and we hope to expand it to provide more services to the user. For example, we hope to provide a hard copy output connection to the Line Processor and allow the user

Donald I. Andrews

to obtain a printout at the same time he uses the workstation for unrelated matters.

715

We expect the microcomputer equipment to become cheaper and faster. These developments will allow the Line Processor and similar devices to have more capabilities. It appears that such devices, if properly designed, could tend to reduce software problems and expenses.

716

A growing trend, brought about by computer networks and interconnections of various kinds, is to divide workloads and define appropriate interfaces. With this method, the computing takes place over several processors, hopefully each best suited to its workload. We have accomplished that with the Line Processor. We are in the infancy of a "distributed computing" era in which microcomputer devices such as the Line Processor will clearly have a growing role.

717

Donald I. Andrews

ACKNOWLEDGEMENTS

7j

The development of the Line Processor has been a team effort by several members of ARC. The hardware design and prototype construction were done by Martin Hardy and Rodney Bondurant. Charles Irby and Kenneth Victor programmed NLS to use Line Processor workstations and modified our TENEX for the Input protocol. The design of the display manipulation protocol and the micro programming were done by the author. Dr. Richard Watson oversaw the effort and was a primary source of useful criticisms during the writing of this paper. We all owe many thanks to Delta Data Systems for extended use of a Delta 5200 video display demonstrator during development of the Line Processor.

7j1

The work reported here is currently supported primarily by the Advanced Research Projects Agency of the Department of Defense, and also by the Rome Air Development Center of the Air Force and by the Office of Naval Research.

7j2

Donald I. Andrews

APPENDIX: Line Processor Protocol and Operation

7k

The Display Manipulation protocol calls for the transmission of unescorted characters and short command strings. Command strings begin with an escape character: we use 34 octal. Characters within the command strings are seven bit ASCII printable characters. Sending printable characters, where possible, rather than control characters makes debugging and troubleshooting much less painful.

7k1

The Line Processor operates in one of two modes. One mode simulates a teletypewriter. The other is the normal display mode that allows NLS display manipulation.

7k2

The mode is specified by mode set commands from the main computer. The Line Processor responds to the "enter display mode" command (also called the interrogate command) by sending a string in protocol format that informs the main computer of the display screen size, length of time it takes to delete a line and the baud rate; the last two parameters are important for timing considerations.

7k3

To make the displays function correctly, it is necessary to send the proper number of "padding" or null characters while the display performs involved functions, like delete line and clear screen.

7k4

The microcomputer is programmed to send the padding characters, but it has limited buffer space and must receive pads from the main computer as well. It would be fine to have

Donald I. Andrews

the main computer refrain from sending anything in most applications, but our users will frequently be connected by way of a computer network. In such an environment, the only way to ensure the proper timing is to send the appropriate number of padding characters.

7k5

The number of padding characters that need to be sent depends on the length of time it takes the display to perform the function and the baud rate going into the Line Processor. So, we have installed a baud rate switch on the Line Processor that can be read by the microcomputer.

7k6

Hence, when responding to the "enter display mode" command, the microcomputer includes the baud rate setting along with the other terminal parameters. From the baud rate and the delete time parameter, the applications program can compute the number of pads to be sent.

7k7

Display Manipulation Protocol Primitives

7k8

POSITION CURSOR (X, Y)

7k8a

This command positions the cursor to the designated spot and stops the mouse tracking process. Any subsequent unescorted characters are written on the display starting at the cursor position.

7k8a1

RESUME TRACKING

7k8b

This command is used after positioning the cursor and writing a string, to start the mouse tracking process again.

7k8b1

Donald I. Andrews

Note: A string is written by sending: POSITION CURSOR;
 (the string); RESUME TRACKING. 7k8b2

DELETE LINE 7k8c

The line at which the cursor has been positioned, is
 deleted. 7k8c1

INSERT LINE 7k8d

A new line is inserted after the line on which the cursor
 was positioned. 7k8d1

CLEAR SCREEN 7k8e

The entire screen is cleared. 7k8e1

BEGIN STANDOUT MODE 7k8f

All text written on the screen subsequent to this command
 will be in "standout mode." 7k8f1

END STANDOUT MODE 7k8g

This returns the text writing mode to normal. 7k8g1

RESET 7k8h

This command resets the line processor to normal mode and
 clears the screen. 7k8h1

WRITE A STRING OF BLANKS (N=number of blanks) 7k8i

It is useful to be able to clear a short area of the
 screen with this command. 7k8i1

PUSH BUG SELECTION (X, Y) 7k8j

The coordinates for the bug selection mark are pushed on
 a stack, and the indicated character is marked. 7k8j1

POP BUG SELECTION 7k8k

Donald I. Andrews

The top entry on the bug selection stack is removed and
the mark at the corresponding screen position is removed. 7k8k1

SPECIFY TTY-SIMULATION WINDOW (Y1, Y2) 7k8l

Y1 and Y2 specify the top and bottom line for the
scrolling window. Any subsequent TTY-type output will be
scrolled in this window. 7k8l1

ENTER DISPLAY MODE (INTERROGATE) 7k8m

ENTER NORMAL MODE 7k8n

Donald I. Andrews

REFERENCES

- 71
1. D. C. Engelbart, R. W. Watson and J. C. Norton,
 "The Augmented Knowledge Workshop,"
 AFIPS Proceedings National Computer Conference, June 1973.
 (SRI Catalog Item 14724) 711
 2. C. H. Irby,
 "Display Techniques for Interactive Text Manipulation,"
 prepared for the Proceedings of the National Computer
 Conference, May 1974.
 (SRI-ARC Catalog Item 20183) 712
 3. L. G. Roberts and B. D. Wessler,
 "The ARPA Network,"
 Advanced Research Projects Agency, Information Processing
 Techniques
 Washington D. C., May 1971.
 (SRI-ARC Catalog Item 7750) 713
 4. D. G. Bobrow, et. al.,
 "TENEX, A Paged Time Sharing System for the PDP-10,"
 presented at ACM Symposium on Operating Systems Principles,
 18-20 October 1971
 Bolt Beranek and Newman Inc., 15 August 1971.
 (SRI-ARC Catalog Item 7736) 714
 5. W. K. English, D. C. Engelbart and M. A. Berman,
 "Display-Selection Techniques for Text Manipulation,"
 IEEE Transactions on Human Factors in Electronics

Donald I. Andrews

Vol HFE-8 Number 1, pp 5-15 March 1967.

(SRI-ARC Catalog Item 9694)

715

6. D. C. Engelbart,

"Design Considerations for Knowledge Workshop Terminals,"

AFIPS Proceedings National Computer Conference, June 1973.

(SRI-ARC Catalog Item 14851)

716

7. D. C. Engelbart and W. K. English,

"A Research Center for Augmenting Human Intellect,"

AFIPS Conference Proceedings, Vol 33, 1968.

(SRI-ARC Catalog Item 3954)

717

8. M. E. Hardy,

"Micro Processor Technology in the Design of Terminal
Systems,"

under preparation for the Proceedings of the IEEE COMPCON,
1974.

(SRI-ARC Catalog Item 20185)

718

Line Processor: A Device for Amplification of Display Terminal
Capabilities for Text Manipulation

(J20184) 28-NOV-73 18:15; Title: Author(s): Don I. Andrews/DIA;
Distribution: /SRI-ARC; Sub-Collections: SRI-ARC; Clerk: DIA;
Origin: <ANDREWS>PAPER.NLS;26, 14-NOV-73 10:42 DIA ;

20184 Distribution

Donald C. (Smokey) Wallace, Richard W. Watson, Don I. Andrews,
James H. Bair, A. Jim Blum, Meredith(Reddy) E. Dively, Jeanne M.
Leavitt, Rodney A. Bondurant, Jeanne M. Beck, Mark Alexander Beach,
Judy D. Cooke, Marcia Lynn Keeney, Carol B. Guilbault, Susan R. Lee,
Elizabeth K. Michael, Charles F. Dornbush, Elizabeth J. (Jake)
Feinler, Kirk E. Kelley, N. Dean Meyer, James E. (Jim) White, Diane
S. Kaye, Paul Rech, Michael D. Kudlick, Ferg R. Ferguson, Douglas C.
Engelbart, Beauregard A. Hardeman, Martin E. Hardy, J. D. Hopper,
Charles H. Irby, Mil E. Jernigan, Harvey G. Lehtman, Jeanne B. North,
James C. Norton, Jeffrey C. Peters, Jake Ratliff, Edwin K. Van De
Riet, Dirk H. Van Nouhuys, Kenneth E. (Ken) Victor