

UNED 1564



Bell Laboratories

subject: An Extended UNIX Text Editor -
ex
Case: 38730-357

date: July 31, 1980
from: E. S. Dattatreya
WH 9311
2A-256 x2542

ABSTRACT

This memorandum describes an extension to the UNIX text editor, ed. The editor described, called ex, can display line numbers, provide automatic indentation, can display a screen full of text, and incorporates some nice PSS features. It is available on the Whippany Computer Center UNIX installations WHUXB, WHUXC, WHUXD and WHUXE.



Bell Laboratories

subject: An extended UNIX Text Editor -
ex
Case: 38730-357

date: July 31, 1980

from: E. S. Dattatreya
WH 9311
2A-256 x2542

MEMORANDUM FOR FILE

1. Introduction

The standard UNIX text editor, ed, [Reference 1] is simple and efficient. However, it lacks some important features such as displaying line numbers, etc. An extended text editor written by the author, ex, is now available on the Whippany Computer Center UNIX installations WHUXB, WHUXC, WHUXD and WHUXE. ex resides in the /usr/sbin directory, and thus can be easily accessed by including this directory in the PATH which is searched by the shell. ex has been in use for over two years by the author and several others. It is being announced now so that others in the local UNIX user community who like the nice features of ex can make use of it.

ex has, as a subset, the commands and features of ed. All of the additional features of ex can be turned off optionally, so that it can be used as though it were ed until the new features are needed.

2. ex Usage

The extended editor is invoked as follows:

```
ex [file1 [file2 [file3 ...]]]
```

where, the optional list of files is referred to as the argument list. The editor enters the edit mode, and begins editing file1. Below are described the additional commands available in ex. Some of the commands take addresses, and the default addresses are shown in parenthesis.

Oletter

The Options command sets the required options depending on the letter, as below:

On

Alternately turns on and off the error message mode. When the mode is on, a short error message is printed when an error occurs. When the mode is off, just a '?' is printed.

Oi

Alternately turns on and off the LINE NUMBERS ON INPUT mode. When the mode is on, in the input mode (i.e., entered by the a, c and i editor commands) each input line is prompted by the corresponding line number, somewhat like the TSO editor.

Om

Alternately turns on and off the MESSAGE mode. When the mode is on, short messages are printed whenever the editor changes from input mode to edit mode or from edit mode to input mode.

On

Alternately turns on and off the LINE NUMBERS mode. When the mode is on, line numbers are displayed whenever a line is printed. This mode can be temporarily reversed by the L command (see below).

Op

Alternately turns on and off the PROMPT mode. In the on state, the editor will prompt with '>' whenever it is waiting for input.

Ot

This option can be used to ease the preparation of structured program text. It alternately turns on and off the AUTO TABS mode. In the on state, in the input mode (via the a, c and i commands), the editor looks at the number of tab characters at the beginning of the previous line added, and automatically adds these characters at the beginning of the next line, and aligns the cursor at the proper level of indentation. For every '<' entered at the beginning of an input line, the cursor backs up one tab stop.

Ov

Alternately turns on and off the AUTO VERIFY mode. When the mode is on, the line that is changed because of an s or u command is printed. The current line is printed after a d command.

(...)L

The List command temporarily reverses the LINE NUMBERS mode, i.e., prints the addressed lines without line numbers when the LINE NUMBERS mode is on and with line numbers when the LINE NUMBERS mode is off.

(.)zchar

The z command prints a screen full of lines. If char is absent, this command prints the next 20 lines (or upto the last line in the buffer). If char is '.', it prints the previous 9 lines, the current line and the next 10 lines. If char is '-', it prints the previous 20 lines. In all cases the buffer boundaries are respected. For example, 10z- prints the first 10 lines, and \$z prints only the last line.

n

The next command edits the next file in the argument list. If there is no next file, a message is printed. It is possible to rewind the argument list using the B command.

b

The backup command rewinds the argument list. Editing of filel is restarted.

All of the current option settings are saved when the editing session is terminated, and are restored at the next editing session. The editor creates an options file .exopt in the login directory of the user to save the options, but this is transparent to the user. The editor includes PSS recommendations such as positive toned "system messages" instead of negative toned "error messages". Other editors are available (Reference 2 and others) which are capable of screen editing and/or provide features similar to ex, but the advantage of ex appears to be that it is only slightly larger than ed, and is as efficient.

An editor session is attached to illustrate the use of the new commands. The PROMPT mode has been turned on so that it is easy to tell the lines that were typed in from those that the editor prints.



WH-9311-ESD-esd

E. S. Dattatreya

Atts.

See next page

Atts.
Editor session
References

Copy (with att.) to
All Supervision in Centers 931 and 932

REFERENCES

- [1] Kernighan, B. W., "A Tutorial Introduction to the UNIX Text Editor," UNIX Library number 1000.
- [2] Joy, W., "An Introduction to Display Editing with Vi," Computer Science Division, University of California, August 9, 1979.

ex.example ex.abs a.c ← we are editing 3 files
ex.example
file does not exist ← message from ex

END - INPUT:

1 > This is a session with the ex editor
2 > to show how the ex commands are used.
3 > . ← to terminate input mode (as in ed)

EDIT:

>lz

1 This is a session with the ex editor
2 to show how the ex commands are used.

2 to show how the ex commands are used.

>2-
1 This is a session with the ex editor
2 to show how the ex commands are used.

>Op ← turns off PROMPT

Op ← turns on PROMPT

>Oi
NO LINE NUMBERS ON INPUT ← changing options.

>a

APPEND - INPUT:

>In the off mode, the l; line numbers do not appear in the
input mode.

EDIT:

>-

3 In the off mode, the l; line numbers do not appear in the
>s;/// 3 In the off mode, the line numbers do not appear in the ← line is auto-matically printed

>u 3 In the off mode, the l; line numbers do not appear in the

>s;/// 3 In the off mode, the line numbers do not appear in the

>ov
NO AUTOVERIFY
← note no line printed after the U command because we turned off VERIFY.

3 In the off mode, the l; line numbers do not appear in the

>s;///

>p

3 In the off mode, the line numbers do not appear in the
>l,\$L ← to temporarily print without line numbers. NOTE: have to use Capital L.

This is a session with the ex editor
to show how the ex commands are used.

In the off mode, the line numbers do not appear in the
input mode.

>on

NO LINE NUMBERS

>z

3 is a session with the ex editor
to show how the ex commands are used.

In the off mode, the line numbers do not appear in the
input mode.

>

```
/* In this example we will show how to use the AUTO TAB feature */
```

```
>
>,e
main()
>{
>int i, j, k, l;
>
>for (i = 1; i <= 100; ++i)
>{
>    j = i + 1;
>    k = i + 2;
>
>    if (k > i)
>    {
>        k = j;
>    }
>    l = k * j;
>    printf("%d", l);
>
>    <
>
>.
EDIT:
>z+
```

— : tab by ex

mm : tab by user

← here the "<" removes one tab.
can use multiple '<'s.

```
/* In this example we will show how to use the AUTO TAB feature */
```

```
main()
{
    int i, j, k, l;
    for (i = 1; i <= 100; ++i)
    {
        j = i + 1;
        k = i + 2;

        if (k > i)
        {
            k = j;
        }
        l = k * j;
        printf("%d", l);
    }
}
```

compare This with the
above.

```
>1,$L
```

```
1 /* In this example we will show how to use the AUTO TAB feature */
2
3 main()
4 {
5     int i, j, k, l;
6
7     for (i = 1; i <= 100; ++i)
8     {
9         j = i + 1;
10        k = i + 2;
11
12        if (k > i)
13        {
14            k = j;
15        }
16        l = k * j;
17        printf("%d", l);
18    }
19 }
```


Printing with line numbers.

>p
the off mode, the line numbers do not appear in the
input mode.

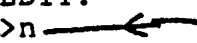
>a
>Note that now there is no prompt with "APPEND - INPUT:" message.
>Nor is there a "EDIT:" message on return to edit mode.
>.

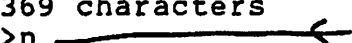
>om
FULL MESSAGES

A: END - INPUT:
>The message mode has been turned ON.


>EDIT:
>oh
>  no error message, just '?'.
?
>oh
>
?
line out of range

>END - INPUT:
>: previous example shows how to turn on/off the error
>message mode.

>.
EDIT:
>n  move to next file.
?
warning: expecting 'w'

>w
9 lines
369 characters
>n  move to file 2.

Editing ex.abs
9 lines
495 characters

>p
.AE
>b  rewind to file 1.

Editing ex.example.
9 lines
369 characters

Editing ex.abs
12 lines
495 characters

Editing a.c
?a.c
file does not exist
>a
APPEND - INPUT:
>