

## Editing

Our editor of choice is “vi” (pronounced vee-eye). We also have an “emacs” and of course “ed”. There is a text editor inside both OpenWindows and Sunview. For a “vi” tutorial, enter “vitutor”. Other starting points would be “man ed” , “man ex”, “man emacs” and/or “man vi”, plus books too numerous to mention here. The SunOS man pages for ex, ed and vi are not useful as references, but the Solaris2 man pages do cover the editing commands.

We have Word Perfect available from OpenWindows . Enter “xwp” from a window. There is no manual entry for “xwp” although it does have its own idea of help features built in. Nedit is another X windows editor that provides mouse based editing for plain text files and on line help.

## Vi Editor

To use vi from a UNIX command line, type:

```
vi filename
```

There are two typing modes. Initially, when you enter a vi, you are in the input mode. To edit, the user enters control mode by typing ESC and moves the cursor to the point needing correction and then inserts or deletes characters or words as needed. Most control commands accept an optional repeat count prior to the command. Here are some of the key and their combinations and the effects they produce when pressed while in control mode (^ means pressing Ctrl together with the letter key specified):

### To MOVE around in a file currently being edited:

^D	Move DOWN one half-screen (depress the control key and type d)
^U	Move UP one half-screen (depress the control key and type u)
^F	Move FORWARD one full-screen
^B	Move BACKWARD one full-screen
^E	Move the window down one line without moving cursor
^Y	Move the window up one line without moving cursor
z<RETURN>	Position the current line to top of window
z	Position the current line to middle of window
z-	Position the current line to bottom of window
^G	Show status of current file
^L	Refresh screen
h	Move left one character
j	Move down one line
k	Move up one line

l	Move right one character
w	Move to the beginning of the next WORD
e	Move to the END of the current word
b	Move BACK to the beginning to the previous word
\$	Move to the end of the line
^	Move to the first non-white character on the line
0	Move to the first column on the line (column zero)
#	Move to an exact column on the line (column #) e.g. 5  12
H	Move to the HIGHEST position in the window
M	Move to the MIDDLE position in the window
L	Move to the LOWEST position in the window
{	(left brace) Move to the beginning of a paragraph
}	(right brace) Move to the end of a paragraph
(	(left paren) Move to the beginning of a sentence
)	(right paren) Move to the beginning of the next sentence
[[	Move to the beginning of a section
]]	Move to the end of a section

### To EDIT text:

dd	DELETE one line
x	X-OUT one character
u	UNDO last change
o	OPEN a line for inserting text
i	INSERT starting at the cursor
O	OPEN a line above the cursor
I	INSERT starting before the cursor
l	INSERT at the beginning of the line
a	APPEND starting after the cursor
A	APPEND at the end of the line
J	JOIN two lines
#s	SUBSTITUTE for # characters
#S	SUBSTITUTE for # whole lines
r	REPLACE character
R	enter over-type mode
f char	FIND the next occurrence of char on the line
t char	Move ‘TIL the next occurrence of char on the line
F char	FIND the previous occurrence of char on the line
/string	Find string looking forward
?string	Find string looking backward

the line	T char	Move ‘TIL the previous occurrence of char on
	;	Repeat the last f, t, F, or T
	,	Reverse the last f, t, F, or T
	%	Show matching () or {} or []
	.	Repeat the last edit command
	m char	MARK this location and name it char
	‘ char	(quote character) return to line named char
	”	(quote quote) return from last movement
	G	GO to the last line in the file
	#G	GO to line #. (e.g., 3G, 5G, 175G)
	n	Repeat last / or ? command
	N	Reverse last / or ? command
	~	(tilde) Convert case of current character

### Colon commands

a colon,	:q!<RETURN>	QUIT without saving changes (type
then the letter q,		then an exclamation point, and press
RETURN)		
	:wq<RETURN>	QUIT after saving changes
	:x<RETURN>	QUIT after saving changes
	:w .....	save file while giving it a new name

#### *Set options*

By typing ":set" as one of the colon commands, a user may see all option values available to them through the use of the set command. For example, set number/nonnumber toggles between line numbering mode and regular mode.

## Pico

pico - simple text editor in the style of the Pine Composer  
 pico [options] [file]Pico is a simple, display-oriented text editor based on the Pine mail system composer.

As with Pine, commands are displayed at the bottom of the screen, and context-sensitive help is provided. As characters are typed they are immediately inserted into the text. Editing commands are entered using control-key combinations.

The editor has five basic features: paragraph justification, case-insensitive searching, block cut/paste, a spelling checker, and a file browser.

Unjustification can be done immediately after justification using the control-U key combination.

String searches are not sensitive to case. A search begins at the current cursor position and wraps around the end of the text.

Blocks of text can be moved, copied or deleted with creative use of the command for mark (ctrl-underscore), delete (ctrl-k) and undelete (ctrl-u).

The spell checker examines all words in the text. It then offers, in turn, each misspelled word for correction while simultaneously highlighting it in the text.

The file browser is offered as an option in the “Read File” and “Write Out” command prompts. It is intended to help in searching for specific files and navigating directory hierarchies. Several basic file manipulation functions are supported: file renaming, copying, and deletion.

The manner in which lines longer than the display width are dealt is not immediately obvious.

Lines that continue beyond the edge of the display are indicated by a ‘\$’ character at the end of the line. Long lines are scrolled horizontally as the cursor moves through them.

More specific help is available in pico’s online help.