



Basic Emacs Editor Commands

What is Emacs?

Emacs is another editor available in UNIX. Like `vi`, emacs is a screen editor. Unlike `vi`, emacs is not an insertion mode editor, meaning that any character typed in emacs is automatically inserted into the file, **unless** it includes a command prefix.

Commands in emacs are either control characters (hold down the `<ctrl>` key while typing another character) or are prefixed by one of a set of reserved characters: `<Esc>` or `<Ctrl>-X`. The `<Esc>` key can be typed by itself (because it really is a character) and then followed by another character; the `<Ctrl>` key must be held down while the next character is being typed. The conventions for describing these characters (since it takes too long to type out the whole thing) are `ESC` means `<Esc>` and `C-` means `<Ctrl>`.

One other distinction between emacs and `vi` is that emacs allows you to edit several files at once. The window for emacs can be divided into several windows, each of which contains a view into a buffer. Each buffer typically corresponds to a different file. Many of the commands listed below are for reading files into new buffers and moving between buffers.

To use emacs on a file, type

```
emacs filename
```

If the file named `filename` exists, then the first screen's worth of the file is displayed; if it doesn't exist, a help message is displayed.

The easiest way to learn emacs is to start it up and go through the on-line tutorial. To access the on-line tutorial, type

```
ESC help-with-tutorial
```

immediately after you have started emacs. The tutorial directs you further in learning the basic commands. One notational point you should know is that the tutorial uses `M-` to mean `ESC`.

To give you a head start, the following table lists the basic commands you need to know to use emacs to edit a file. An asterisk (* or star) to the left of a command indicate it is one to learn immediately.

Help Commands

*	C-h	<i>help-command: first character in lots of useful help commands</i>
*	C-h t	<i>help-with-tutorial: command to run the tutorial</i>
	C-h i	<i>information: describes most of the <code>emacs</code> commands in man style pages</i>
	C-h k	<i>describe-key: tells you what a particular key stroke does</i>
*	C-h a	<i>command-apropos: prompts for a string and then searches for all <code>emacs</code> commands that contains that string</i>
	ESC ?	<i>also does command-apropos</i>
*	C-h ?	<i>help-for-help: describes how to use the help facilities</i>

File Reading and Writing Commands

*	C-x C-f	<i>find-file: first prompts for a filename and then loads that file into a editor buffer of the same name</i>
*	C-x C-s	<i>save-buffer: saves the buffer into the associated filename</i>
	C-x C-w	<i>write-named-file: prompts for a new filename and writes the buffer into it</i>

Cursor/Screen Movement Commands

Depending on the terminal, some of the cursor movement can be handled by the arrow keys.

*	C-a	<i>move cursor to (at) beginning-of-line</i>
	C-e	<i>move cursor to end-of-line</i>
*	C-f	<i>move cursor forward one character</i>
*	C-b	<i>move cursor backward one character</i>
*	C-n	<i>move cursor to next line</i>
*	C-p	<i>move cursor to previous line</i>
	C-v	<i>scroll file forward by one screenful</i>
	ESC v	<i>scroll file backward by one screenful</i>
*	ESC <	<i>go to beginning-of-buffer</i>
*	ESC >	<i>go to end-of-buffer</i>
	ESC f	<i>move cursor forward one word</i>
	ESC b	<i>move cursor backward one word</i>

Copy and Delete Commands

	C-d	<i>delete-char: delete character under cursor</i>
--	------------	--

	ESC d	<i>delete-word: delete from cursor to end of word immediately ahead of the cursor</i>
*	C-k	<i>kill-line: delete the rest of the current line</i>
*	C-@	<i>set-mark-command: mark is used to indicate the beginning of an area of text to be yanked</i>
*	C-w	<i>kill-region: delete the area of text between the mark and the current cursor position</i>
*	C-y	<i>yank: insert at current cursor location whatever was most recently deleted</i>
	ESC w	<i>copy-region-as-kill: copy area between mark and cursor into kill-buffer so that it can be yanked into someplace else</i>

Search Commands

*	C-s	<i>isearch-forward: prompts for text string and then searches from the current cursor position forwards in the buffer</i>
	C-r	<i>isearch-backward: like isearch-forward, but searches from the current cursor position to end of buffer for text string</i>
	ESC %	<i>query-replace: prompts for a search string and a string with which to replace the search string</i>

Window and Buffer Commands

	C-x 0	<i>zero-window: deletes current window</i>
	C-x 2	<i>double-window: splits current window into two parts, allowing you to edit at two different locations in the same file or permitting you to view two different files at the same time</i>
	C-x b	<i>switch-to-buffer: display a different buffer on the screen</i>
	C-x o	<i>other-window: move the cursor to the other window (assuming that you have two windows/buffers open at once)</i>
*	C-x C-b	<i>list-buffers: lists those buffers currently loaded into emacs</i>

Exiting Emacs, Fixing Mistakes and Other Important Stuff

*	C-x C-c	<i>save-buffers-kill-emacs: when you are finished editing, to save the edited but unsaved buffers and to return you to the UNIX prompt</i>
*	C-g	<i>keyboard-quit: if while typing a command you make a mistake and want to stop, this aborts a command in progress</i>
	C-u	<i>universal-argument: if you want to do a command several times, type this command followed by a number (for the number of times)</i>

		<i>followed by the command you wish repeated</i>
*	C-x u	<i>undo</i> : undoes the last command typed, in case you made a mistake
*	ESC x	<i>execute-extended-command</i> : prompts for the name of an emacs command; allows you to execute a command if you know roughly what it is called but cannot remember the key strokes for it

Other Emacs Bindings

Emacs has many other useful commands. As you get more proficient at it, try listing the key bindings (**ESC-x** for ***describe-bindings***) to find other commands. Also note that emacs has what is called ***escape-completion***: as you type in commands or filenames, you can type a space at any time and emacs will either fill out the rest (if there is only one possible completion) or it will list the possible completions in a new window.

A Web page with a complete list of [emacs bindings](#) is also available.

Comments: schauble@cs.colostate.edu

Copyright © 1998: Colorado State University, CS Department. All rights reserved.
