CS540: Artificial Intelligence

Computer Science Department, Colorado State University - Spring 1997

Basic Emacs Editor Commands

1 Overview of 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 [CONTROL] key while typing another character) or are prefixed by one of a set of reserved characters: [ESC] or [CONTROL]-X. [ESC] can be typed by itself (because it really is a character) and then followed by another character; [CONTROL] 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 [CONTROL].

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.

2 Using Emacs, Getting Started

To use emacs on a file, type "emacs filename". If the filename exists, then the first screen's worth of the page will be displayed; if it doesn't exist, a help message will be 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 tutoiral will direct 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 lists the basic commands you will need to know to use emacs to edit a file. ! to the left of the commands indicate those to learn immediately.

Help Commands

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

File Reading and Writing Commands

!	C-x C-f	find-file, first prompts for a filename and then loads that file into a editor buffer of the same name
!	C-x C-s	save-buffer, saves the buffer into the associated filename
	C-x C-w	Write named file, prompts for a new filename and writes the buffer into it

Cursor/Screen Movement Commands

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

!	C-a	put cursor at beginning-of-line
	C-e	put cursor at end-of-line
!	C-b	go backward one char
!	C-f	go forward one char
!	C-n	go to next line
!	C-p	go to previous-line
	C-v	scroll-up by one screenful
	ESC v	scroll back by one screenful
!	ESC <	go to beginning-of-buffer
!	ESC >	go to end-of-buffer
	ESC b	go backward one word
	ESC f	go forward one word

Copy and Delete Commands

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

Search Commands

	C-r	isearch-backward, prompts for text string and then searches from the cursor position backwards in the buffer
!	C-s	isearch-forward, like isearch-backward, but searches from current position to end of buffer for text string
	ESC %	query-replace, prompts for a search string and a string with which to replace the search string

Window and Buffer Commands

	C-x 0	deletes current window
	C-x 2	splits current window into two parts, so you can edit at two different locations in the same file or so that you can view two different files at the same time
	C-x b	switch-to-buffer, display a different buffer on the screen
	C-x o	move the cursor to the other window (assuming that you have two windows/buffers open at once)
!	C-x C-b	list-buffers currently loaded into emacs

Exiting Emacs, Fixing Mistakes and Other Important Stuff

!	C-x C-c	save-buffers-kill-emacs, when you are finished editting, it saves editted but unsaved buffers and then returns you to unix
!	C-g	keyboard-quit, if while typing a command you make a mistake and want to stop, this aborts commands in progress
	C-u	universal-argument, if you want to do a command several times type this command followed by a number (for the number of times) followed by the command you wish repeated
!	C-x u	undoes the last command typed, in case you made a mistake
!	ESC x	execute-extended-command, prompts for the name of an emacs command, allows you to execute commands if you know roughly what it is called but cannot remember the key strokes for it

Emacs has many other useful commands. As you get more proficient at it, try listing the key bindings (ESC-x describe-bindings) to find other commands. Also note that emacs has what is called escape-completion, meaning that 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.