Information Technology
Rice University
July 10, 2000
Document No. UNIX 5.01
GNU Emacs Reference Card

ENTERING AND EXITING GNU Emacs

GNU Emacs	start GNU Emacs
C-z	temporarily suspend GNU Emacs
C-x C-c	quit GNU Emacs

KEYS AND THE KEYBOARD

There are a number of special keys that are used with GNU Emacs. Most commands require a combination of strokes. These are the basic key strikes that are required.

C-x	Simultaneously press CONTROL and x. This is used to begin many different commands.
META or M-	Special Emacs key used to begin many commands. This key is pressed. and then released before typing the next character. On most systems, this is the ESC key, but it can be changed.
M-x	Press META and then hit x
.	
C-u *n*	The prefix used to designate the number of times you want the command executed (over one), where *n* is the number of times. This is known as the argument of a command. The argument comes before the command.
M*n* command	This sequence is also used to repeat a command, by typing META and then *n*, where *n* is the number of times you want the command that follows to be executed.

A buffer is somewhat analogous to a file, but it refers more specifically to a particular editing session. Using GNU Emacs, you can open many buffers because they are separated from the files from which they came.

TRANSPOSING

C-t	transpose character
M-t	transpose words
M-x C-t	transpose lines
M-C-t	transpose sexp

FILES

C-x C-f	read a file into Emacs
C-x C-s	save a file back to disk
C-x i	insert contents of another file into this buffer
C-x C-v	replace this file with the file you really want
C-x C-w	write buffer to a specified file
C-x d	run Dired, the directory editor

PRINTING

You can print an entire file (buffer), or you can print a particular region defined by mark and point.

lpr region
lpr buffer

GETTING HELP

The Help system is simple. Type C-h and follow the directions. If you are a first time user, type C-h t for a tutorial (This card assumes you know the tutorial).

C-x 1	get rid of Help (or any other) window
a.bd ESC C-v!	scroll Help window
C-h a	apropos: show commands matching a string
C-h a	show the function a key runs
C-h f	describe a function
C-h m	get current mode-specific information

ERROR RECOVERY

C-g	abort partially typed or executing command
M-x recover-file	recover a file lost by a system crash
C-x or C-_	undo an unwanted change
M-x revert-buffer	restores a buffer to its original contents
C-l	redraw screen with cursor line in the center of buffer

INCREMENTAL SEARCH

C-s	search forward
C-r	search backward
M-C-s	regular expression match

Use C-s or C-r again to repeat the searching either direction

ESC	exit incremental search
DEL	undo effect of last character
C-g	abort current search (also the general abort command)

If GNU Emacs is still searching, C-g will cancel the part of the search not done, otherwise it aborts the entire search.

MOTION

Cursor Motion:

backward	forward	entity to move over
C-b	C-f	character
M-b	M-f	word
C-p	C-n	line
C-a	C-e	go to beginning (or end) of sentence
M-[	M-]	paragraph
C-x [	C-x]	page
M-C-b	M-C-f	sexp (for programming languages)
M-C-a	M-C-e	function
M-<	M->	go to buffer beginning (or end)

KILLING AND DELETING

backward	forward	entity to kill
DEL	C-d	character (delete, not kill)
M-DEL	M-d	word
M-0 C-k	C-k	line (to end of)
C-x DEL	M-k	sentence
M--M-C-k	M-C-k	sexp

C-w	kill region
M-z char	kill to next occurrence of char
C-y	yank back last thing killed
M-y	replace last yank with previous kill

MARKING

Creates a region for a command to operate on. One end is bounded by the mark, the other by the point. The mark remains there until you reset it, but GNU Emacs cannot show you where it is.

C-@ or C-SPACE	set mark here
C-x C-x	exchange point and mark
M-@	set mark arg words away
M-h	mark paragraph
C-x C-p	mark page
M-C-@	mark sexp
M-C-h	mark function
C-x h	mark entire buffer

BUFFERS

C-x b	select another buffer
C-x C-b	list all buffers
C-x k	kill a buffer

QUERY REPLACE

You should use this instead of REPLACE when you want to replace some occurrences of an expression but not all of them.

M-x query replace *word newword*
M-% *string newstring* interactively replace a lost string
M-x query-replace-*regexp* using regular expressions

Valid responses in query-replace mode are

SPACE replace this one, go on to next and pause
replace this one, don't move
DEL skip to next without replacing
replace all remaining matches
back up to the previous match
ESC exit query-replace
C-r enter recursive edit (C-M-c) to exit

KEYBOARD MACROS

You can customize keyboard macros within a file to abbreviate a sequence of keys or commands that you use frequently. This is how you define them.

C-x (start defining keyboard macro
C-x) end keyboard macro definition
C-x e execute last-defined keyboard macro
C-u C-x (append to last keyboard macro
M-x name-last-kbd-macro name last keyboard macro
M-x insert-kbd-macro insert lisp definition buffer
M-x *name* to invoke a defined keyboard macro

THE MINIBUFFER

This is the bottom line in your GNU Emacs window. It echoes all commands.

The following keys are defined in the minibuffer.

TAB complete as much as possible
SPACE complete up to one word
RET complete and execute
show possible completions
C-g abort command

Type C-x ESC to edit and repeat the last command that used the minibuffer. The following keys are then defined.

M-p previous minibuffer command
M-n next minibuffer command

MULTIPLE WINDOWS

C-x 1 delete all other windows
C-x 0 delete this window
C-x 2 split window in two horizontally
C-x 5 split window in two vertically

M-C-v scroll other window
C-x o with cursor to another window

M-x shrink window shorter
C-x ^ grow window taller
C-x { shrink window narrower
C-x } grow window wider

C-x 4 b select a buffer in other window
C-x 4 f find file in other window
C-x 4 m compose mail in other window
C-x 4 d run Dired in other window
C-x 4 . find tag in other window

FORMATTING

TAB indent cursor line (mode-dependent)
M-C-\ indent region (mode-dependent)
M-C-q indent sexp (mode-dependent)
C-x TAB indent region rigidly arg columns

C-o insert new line after point
M-C-o move rest of line vertically down
C-x C-o delete blank lines around point

M-\ delete all whitespace around point
M-SPC put exactly one space at point

M-q fill paragraph
M-g fill-region
C-x f set fill column
C-x . set prefix each line starts with

SHELLS

M-! find tag
C-u M-. tag find next occurrence of tag
M-x visit-tags-table specify a new tags file

M-x tags-search reg exp search on all files in tags table
M-x tags-query-replace query replace on all the files
M-, continue last tags search or query-replace

TAGS

M-. *tag* find tag
C-u M-. *tag* find next

CASE CHANGE

note: M-- means press META, release and press the minus (-) key.

last word	next word	case
M-- M-u	M-u	uppercase
M-- M-l	M-l	lowercase
M-- M-c	M-c	capitalize

C-x C-u uppercase region
C-x C-l lowercase region
M-x capitalize region

SPELLING CHECK

M-$ check spelling of current word
M-x spell-region check spelling of all words in region
M-x spell-buffer check spelling of entire buffer

If the word is incorrect, GNU Emacs will ask you to edit it, and will then do a query replace (so you must use the query replace command to actually fix your mistakes).

REGULAR EXPRESSIONS

The following have special meaning inside a regular expression.

. (dot) any single character
* zero or more repeats
+ one or more repeats
? zero or one repeat
[...] any character in set
[^...] any character not in set
^ beginning of line
$ end of line
\ quote a special character c
\| alternative or
\(...\) grouping
\n nth group
\` beginning of buffer
\' end of buffer
\b word break
\B not beginning or end of word
\< beginning of word
\> end of word
\w any word-syntax character
\W any non-word-syntax character
\s c character with syntax c
\S c character with syntax not c