Introduction to the Emacs Text Editor

Platform: UNIX

Level of Difficulty: Beginner

Introduction

A text editor is a program that manipulates plain text (ASCII). Most text editors are full screen editors (they display a full screen of text) as opposed to line editors (which display one line at a time). Emacs is a full screen text editor supported by RUCS. On RUCS UNIX machines the default version of Emacs is the GNU software release of Emacs.

Documentation Notes

<CTRL> means press the control key and, as you are holding it down, type the character that follows. For example, **<CTRL>m** means that you press the control key and, while holding it down, press the **m** key. The control key is usually labeled **Ctrl** or **Control**; some documents show this as a caret (**^**), the Emacs tutorial indicates control with **C-**.

<ESC> means press and release the escape key, then type the character that follows. For example, **<ESC> m** means that you press and release the escape key, and then press the **m** key. The escape key is usually labeled **Escape** or **Esc**; the Emacs tutorial indicates escape with **M-** or **META**.

<CR> means type the Return key, which is usually labeled **Return**, **Carriage Return**, or **Enter**.

<DEL> means type the Delete key, which is usually labeled **Delete**, **Backspace**, or **Rubout**.

<SPACE> means press the space bar. To make the commands in this document more readable, there are spaces between some of the control characters. These spaces are not to be entered: if a space is required in a command it will be shown as **<SPACE>**.

Naming Conventions

Commands to be entered on the keyboard are shown in the "**typewriter**" type; explanations are shown in this type. A common naming convention for files that is acceptable on most machines is to have the main part of the name deal with the subject of the file, then a period (.), and then an extension that describes the type of file. Emacs is intelligent enough to interpret some extensions and provide an editing environment (called a

mode) that is appropriate to that type of file. Files with the extension .txt will be placed in the "text mode", those with .f will be placed in "fortran mode", etc. For more information on modes see the RUCS document Emacs Modes. For a description of the current mode issue the command **<CTRL>h m**

Emacs Concepts

In Emacs a buffer is an area of memory where editing is done. These edits are not reflected in files until that buffer is saved into a file. In Emacs white space is blank lines, spaces, or tabs. Emacs paragraphs are separated by indented lines or blank lines. Emacs sentences are terminated by a paragraph separator or one of the following punctuation marks followed by TWO spaces; a period, an exclamation point, or a question mark.

Getting Started

To start Emacs type `emacs<SPACE>filename` or `emacs`. If you do not give it the name of a file to bring into a buffer, you will have a scratch window that will not be saved automatically when you exit Emacs.

Exiting

When you enter:	**Emacs will:**
<CRTL>x <CRTL>c	exit Emacs. If you have not saved all the buffers you have changed, it will ask if you want to save those buffers before you exit. NOTE: if a buffer is not associated with a file, this exit will not ask to save your work.
<CRTL>z	suspend Emacs and return to command level. In UNIX you can have multiple sessions suspended and resume one with either the **%** or **fg** command.
<CTRL>g	exit currently executing command.

Help in Emacs

These are a few of the informational commands available within Emacs.

When you enter:	**Emacs will:**
<CTRL>h	start the help command.
<CTRL>h a	search for commands that contain the supplied keyword.

<CTRL>h c	when followed by a command keystroke, show command name.
<CTRL>h i	start the "info" information menu of online Emacs documentation. (This is NOT part of the Rutgers INFO system.)
<CTRL>h k	when followed by a command keystroke, describe the command.
<CTRL>h t	start the Emacs tutorial.—Recommended method for learning Emacs.
<CTRL>h ?	list the help sub-commands.
<CTRL>h <CTRL>h	list and describe the help subcommands.

Cursor Movement

Moving your cursor allows you to select where in the displayed text the current action is to take place. On some screens the cursor is a flashing underscore character (_), on others it is a reverse video rectangle (n)

When you enter:	**Emacs will:**
<CTRL>f	move the cursor forward one character.
<CTRL>b	move the cursor backward one character.
<ESC> f	move the cursor forward one word.
<ESC> b	move the cursor backward one word.
<ESC> e	move the cursor forward one sentence.
<ESC> a	move the cursor backward one sentence.
<ESC>]	move the cursor forward one paragraph.
<ESC> [	move the cursor backward one paragraph.
<CTRL>e	move the cursor to the end of the current line.
<CTRL>a	move the cursor to the beginning of the current line.
<CTRL>p	move the cursor to the previous line.
<CTRL>n	move the cursor to the next line.

Display Control

Controlling the display allows you to select what part of the buffer you are looking at.

When you enter:	**Emacs will:**
<CTRL>l	clear screen and redisplay with the line the cursor is on in the center.
<CTRL>v	scroll to the next screen.
<ESC> v	scroll to the previous screen.
<ESC> <	move the cursor to the beginning of the buffer

<ESC> >	move the cursor to the end of the buffer
<CRTL>x [	move the cursor to the preceeding **<CRTL>l**, or beginning of file.
<CRTL>x]	move the cursor to the next **<CTRL>l**, or end of file.

Text Handling

Changing existing text requires the text handling commands.

When you enter:	**Emacs will:**
<CR>	end current line and start new line
<CTRL>o	open a new line at the cursor.
<CTRL>q	quote insert a control character into the text. The most common control character inserted is **<CTRL>l** which signifies a page break (this is achieved by typing **<CTRL>q <CTRL>l**).
<ESC> q	adjust current paragraph to fill out to the fill column (70 by default).
<DEL>	delete the character to the left of the cursor.
<CTRL>d	delete the character at the cursor.
<ESC> d	kill (delete) from the cursor to the end of the current word
<ESC> <DEL>	kill from the cursor to the beginning of the current word.
<CTRL>u	repeat a command multiple times. To go down 10 lines type: **<CTRL>u 10 <CTRL>n**
<CTRL>k	kill from the cursor to the end of the current line.
<ESC> k	kill from the cursor to the end of the current sentence.
<CTRL>x <DEL>	kill from the cursor to the beginning of the current sentence.
<CTRL>t	transpose the character at the cursor with the one to the left of it
<ESC> t	transpose the words on either side of the cursor.
<CTRL>x <CTRL>t	transpose the current line with the one above, place cursor on the next line.
<CTRL>_	undo the previous action. Repeated undos will undo preceding actions.
<CTRL>@	place a mark at the cursor. A region is the text between a mark and the cursor.
<CTRL><SPACE>	place a mark at the cursor
<CTRL>x <CTRL>x	exchange point (cursor) and mark, allows you to see both ends of a region.
<CTRL>w	kill region.
<ESC> w	copy region.

<CTRL>y	yank back (restore) the most recent deletion done by any of the above kill commands or by a copy.
<ESC> y	yank back (restore) a previous deletion done by any of the above kill commands or by a copy. MUST be preceded by a <CTRL>y.

HINT: an effective way to move whole regions of text from place to place is to set a mark (with **<CTRL>@**) at the top of the region you wish to move, then move the cursor down to the bottom of the region and then delete the region (with **<CTRL>w**). Then move the cursor to the place where you want this region moved to, and restore (yank) the region back in (with **<CTRL>y**).

Reading and Writing Files

When you enter:	**Emacs will:**
<CRTL>x <CRTL>f	find a file for editing, create a new buffer.
<CRTL>x <CRTL>v	visit a file for editing, discard current buffer contents.
<CRTL>x <CRTL>s	save the current buffer into a file.
<CRTL>x s	prompt you to save every modified buffer.
<CRTL>x <CRTL>w	write the current buffer into a file and prompt you for a filename.
<CRTL>x i	prompt you for a file to insert at the cursor.

Searching

To find words and phrases in the text of a file you need to use search commands. The default search commands are incremental searches, i.e., they search for the first character entered then for the first followed by the second character, etc.

When you enter:	**Emacs will:**
<CTRL>s	incrementally search forward from the cursor for the given string.
<CTRL>r	incrementally search backward from the cursor for the given string.

Notes: To end either search, type **<ESC>** or move the cursor with **<CTRL>b**, **<CTRL>f** etc. The last string that was searched for is remembered, type **<CTRL>s** (or **<CTRL>r**) twice to start a search and to search for the previous string.

Replacing Strings

To change one portion of text to another throughout a text manually would often be unfeasible. The following commands automate such modifications.

When you enter:	Emacs will:
<ESC>x replace-string<CR>	Prompts for old-string and new-string, then replaces all occurrences of the old-string with new-string, from the cursor to the end of the buffer.
<ESC> %	operate like replace-string, but will check before doing a replacement. This function (query-replace) uses the following sub-commands:
<SPACE> or y	make current change and go to next match
.	make current change and quit
<DEL> or n	don't make current change and go to next match
<ESC>, <CTRL>g or q	don't make current change and quit

Changing Cases

At various times it is desirable to change the case of portion of text (all upper case, all lower case, capitalize the first letter of a word), the following commands affect the case of text.

When you enter:	Emacs will:
<ESC> l	lowercase all letters from cursor to the end of the current word.
<ESC> u	uppercase all letters from cursor to the end of the current word.
<CTRL>x <CTRL>l	set all letters in the region to lowercase.
<CTRL>x <CTRL>u	set all letters in the region to uppercase.
<ESC> c	capitalizes from cursor to the end of the current word.

Dealing with multiple Windows

There are times when it is necessary to view two or more files (or sections of the same file) at the same time. Emacs allows you to have more than one buffer visible at one time. The following commands are used to work in multiple windows.

When you enter:	Emacs will:
<CTRL>x O	delete the current window (Note: zero, not alphabetic "O").

<CTRL>x 1	delete all other windows but the current one.
<CTRL>x 2	split the current window into two, one over the other.
<CTRL>x 5	split the current window into two, one beside the other.
<CTRL>x o	move cursor to the other window (alphabetic "O" not zero).
<CTRL>x ^	enlarge the current window one line (**^** is the caret character).
<ESC> <CTRL>v	scroll the other window downward.
<CTRL>x 4 f	find a new file in another window, if there is only one window it will be split in two like **<CTRL>x 2**.

HINT: you can use two windows to copy lines from one buffer to another. Split the window, place one buffer in each window, then kill or copy (see Text Handling) the text from one buffer, switch windows, and yank (see Text Handling) the text into the other buffer.

Buffers

Buffers is the term Emacs uses to refer to the portion of memory that has a body of text in it for manipulation. Often buffers contain copies of the contents of specific files. The below commands are the basic commands for working with buffers.

When you enter:	**Emacs will:**
<CTRL>x <CTRL>b	list current buffers.
<CTRL>x b	switch to an existing buffer. You will be prompted for a buffer name.
<CTRL>x 4 b	split the window, prompt for a buffer name, bring that buffer into the second window, and place the cursor in the new buffer.

Keyboard Macros

Keyboard macros are useful for strings of commands that will be repeated many times.

When you enter:	**Emacs will:**
<CTRL>x (	start recording key strokes to build a keyboard macro.
<CTRL>x)	stop recording key strokes to for a keyboard macro.
<CTRL>x e	execute the keyboard macro. Only one keyboard macro can be defined at a time. This is very powerful when used in conjunction with **<CTRL>u**
<CTRL>u	repeats a command. For 10 repetitions, type **<CTRL>u** 10 and then the command.