

Using GNU EMACS for the First Time

October 1, 2006

1 What is EMACS ?

EMACS is an advanced, self-documenting, customizable and extensible real-time display editor. To explain this further, we say it is advanced because it provides facilities beyond that of a standard editor. EMACS can do more than just insertions, deletions and one-character manipulations. Functions such as: filling of text, automatic indentation, viewing multiple files at once, and the concept of dealing with text in terms of “words”, “lines”, “paragraphs” and “sentences” instead of just “characters” make it very powerful and useful. We call it self-documenting because at any time you can type a help command and get help on EMACS, specific commands, a particular key, or even execute an on-line tutorial program. We call it customizable because you can define keys to perform functions to suit your own needs (called Key-Bindings), or you may choose to actually change the way built-in functions behave. EMACS is extensible because new commands can be added to it very easily without changing the rest of it. There are also many extensions available to make programming, text processing and even reading *mail* easier by having EMACS do much of the work for you (such as parenthesis matching, automatic “pretty printing” for programs, and special modes for \TeX and \LaTeX). Lastly, we call it real-time because everything happens on the screen in front of you.

EMACS is available on most Unix systems. ALL of the RCF machines run the same version of GNU EMACS. Even though implementations may be slightly different, the user can always customize it to their own preferences.

2 EMACS Concepts

EMACS can be run on almost any terminal with screen-addressing capabilities (e.g. almost anything except for a Teletype, DECwriter, or other printing terminal). If you are using a terminal emulator on a PC, a DEC VT-100 emulator will work best. Please ask one of the **staff** if you have any questions regarding specific terminal types.

2.1 Buffers

A *buffer* is the basic “holding area” for EMACS. When you edit a file, what actually happens is that EMACS will read the contents of the file into a *buffer* and give it the same name as the file. This is a very important concept to understand, because the contents of the file will NOT be changed until you give EMACS the command to write the *buffer* back out. In general, EMACS is very good about telling you when you have forgotten to write out the contents of a *buffer* you have changed before it lets you exit. *Buffers* can be associated with files, shells (you can actually run system commands from within EMACS), and EMACS functions (such as HELP). There is no practical limit to the number of *buffers* that you can use, and it is very easy to switch between them.

2.2 Windows

A *window* is a piece of the screen that displays a *buffer*. You can have more than one *window* on the screen at once. You can have multiple *windows* each displaying different sections of a single *buffer*. In practical use, you can look at two (or more) files at once, or two (or more) places in the same file at once, and move text between them. The number of *windows* you can have on the screen at once is only limited by the number of lines on the display.

2.3 The *mode-line*

At the bottom of the screen you will see the *mode-line* displayed in either reverse-video, or bold-face depending on the terminal type. This line shows you that you are in EMACS. This is followed by the name of the *buffer* that you are in, the “mode” of EMACS that you are using, and your position in the text. When you create a new file you will be looking at a blank screen.

The position field will say **-ALL-** because you are looking at *all* of the screen. When the *buffer* is larger than one screen, the position field will be expressed as either **-TOP-** (if you are at the beginning), **-BOTTOM-** (if you are at the end), or a percentage of how far you are into the text (if you are somewhere in the middle). This *mode-line* line is very useful. If you look carefully, you will notice that on the left of the line there are some dashes and asterisks. These signify the status of the *buffer*. If this part of the line looks like “- - * * **-Emacs:**” then you have a *modified-buffer* and it needs to be saved. If this part is all dashes “- - - - **-Emacs:**” then no changes have been made since the last time the *buffer* was saved. If you want to edit two files at once, you can have them both up on the screen simultaneously in separate *windows*, and each *window* will have its own *mode-line*.

2.4 Entering Commands

Most EMACS commands are entered by using one or two keystrokes. If you are using a VT-series or Esprit terminal, you can use the default function keypads for most purposes. A copy of these keypad definitions is available from the RCF staff. It is worthwhile to learn how to use the keypads because it will save you a lot of time and typing. To enter commands NOT included on the keypad, use either control-sequences or the *minibuffer*. The three basic command prefixes are:

- *Control* – (**C-**) which means “Hold down **Ctrl** Key while hitting the next key”. For example: The command for transposing two characters is *transpose-chars*, or the keystroke **C-t**. This means “Hold down the **Ctrl** key and press **t**. This will cause the *transpose-chars* command to be executed.
- *Meta* – (**M-**) which means “Press *Meta* key before hitting the next character”. By default the *Meta* key is **ESC** (The escape key, which is F11 on VT-220 terminals, and **ESC** on the Esprit and most other terminals). For example: The command to transpose two words is *transpose-words*, or the keystroke **M-t**. This means “Press the *Meta* key (which is **ESC**) then press **t**. This will cause the *transpose-words* command to be executed.
- *Meta-x* – (**M-x**) which means “Press **ESC**, then press **x**”. This tells EMACS to run a command by name. When you enter this, EMACS

will ask you to fill in the command name at the bottom of the screen. This part of the screen is called the *minibuffer*. After you have entered the command, press the \leftrightarrow (RETURN) key to execute it. For example: Another way to run the *transpose-words* command would be to enter the command **M-x transpose-words** \leftrightarrow . This would be done by pressing **M-x** (press **ESC**, and then press **x**). Then type **transpose-words** \leftrightarrow , and the command will be executed.

3 Running EMACS

To start up EMACS, simply type *emacs* at the system command prompt. If you wish to create a new file, or edit an existing file, you may include the name of that file on the command line. The following example will start EMACS and edit the file *testfile*.

```
% emacs testfile  $\leftrightarrow$ 
```

After this is done, the screen will clear and the cursor will be in the top-left corner of the screen.

3.1 Creating Text

Once EMACS is started, you may begin editing. If you are creating a new file, the *buffer* will be empty and you can begin typing out your text. If you do not give EMACS a filename when it is started, you will be in a *buffer* named ***scratch***. The section on *Reading and Saving Files* describes the command for reading in a file after EMACS is started. By default, EMACS will do automatic “line-wrapping” so that you do not have to press a \leftrightarrow at the end of each line.

3.2 Moving around

The easiest way to move around a document is to use the *next-line*, *previous-line*, *forward-character* and *backward-character* functions to move one line or character at a time. Since this can get a little tedious in a large document, you can also move by word, by line, by sentence, by paragraph or by screen. The VT-series and Esprit keypad diagrams show how the function keys on these terminals are bound by default.

3.3 *Mark and Point*

In EMACS, you use a *mark* and a *point* to delimit specific regions of text for killing (deleting), moving, copying, saving, or indenting. The *mark* is an invisible pointer that you set at the beginning of a region by using the *set-mark* command. The *point* is always located to the left of the current cursor position.

3.4 **Deleting Text**

If you only need to do small deletions, you can use the *delete-previous-character* and *delete-next-character* functions. There are also the *delete-previous-word* and *delete-next-word* commands for deleting words and the *kill-from-start-of-line* and *kill-to-end-of-line* commands for deleting lines. If you need to delete large sections of text or want to move large amounts of text, you should use the *set-mark*, *kill-region* and *yank* functions as follows:

1. *set-mark* – (**C-** or **C-SPC**) Sets a *mark* in the text at the position where you want to begin killing (deleting).
2. Move the cursor to the end of the region that you want killed.
3. *kill-region* – (**C-w**) Kills the region of text between the *mark* and the *point* (to the left of where the cursor is). If you are only deleting text then you are done. If you want to move this region somewhere else, then do the next two steps:
4. Move the cursor to the position at which you want the text inserted. This can be anywhere in the buffer, or you can switch to another buffer if you like.
5. *yank* – (**C-y**) Yanks the region which was killed and inserts it at the current *point*.

3.5 **Reading and Saving Files**

- *find-file* – (**C-x C-f**) This command will prompt for a file name, create a new buffer with the same name as the file, and then read that file into the buffer.

- *insert-file* – (**C-x i**) This command inserts the contents of a file from disk at the pointer location of the current file.
- *save-buffer* – (**C-x C-s**) This command will save the current buffer into a file. When you save a buffer, EMACS will use the current buffer name for the filename.
- *write-buffer* – (**C-x C-w**) This command will ask you to give it a filename, then save the current buffer into that file.

3.6 Error Recovery

There is a way to *undo* commands in EMACS. *undo* (**C-x u**) undoes the last command. Consecutive repetitions of *undo* will *undo* earlier and earlier changes. The limit to the number of changes that you can *undo* varies depending on the complexity of the previous commands, but it is more than sufficient for most purposes.

Another useful command is *revert-buffer* (**M-x revert-buffer**). This command will revert the state of the current buffer back to its original contents. This command will only work on buffers that were loaded from files. When you *revert-buffer* the buffer will be re-loaded from disk with the previous version of the file.

Another problem that may be encountered occurs when EMACS is aborted abnormally. If the system crashes while you are in an editing session, your work is not lost. When you edit a file from a session that was abnormally aborted, EMACS will send a message to the *minibuffer* telling you to use the *recover-file* command. When you invoke the *recover-file* command (**M-x recover-file**), your last editing session will be “replayed” on the screen. This will restore as much of the lost session as possible.

If you make an error entering a command, or want to stop EMACS in the middle of a command, use *abort* (**C-g**). This command will *abort* a partially typed command, or a command that is currently executing.

3.7 Exiting EMACS

You can leave EMACS in two different ways:

- *save-buffers-kill-emacs* – (**C-x C-c**) This command will automatically ask you if you want to save any modified buffers before exiting EMACS.

Answer **yes** or **no** when it asks you about saving each modified buffer.

- *suspend-emacs* – (**C-z**) This command will suspend the current EMACS process in the current state. You may re-attach to it by using a shell command such as `%emacs ↵` in the C-shell.

4 Getting Help

At the beginning of this document we said that EMACS was self-documenting. There are a variety of help facilities available within EMACS. The main help functions and keystrokes to invoke them are:

- *help* – (**C-h**) This command will put you into the *help* facility with no options. This will ask you for an option, or you may enter `?` to get a list of options.
- *info* – (**C-h i**) This will begin the *info* documentation browsing system. This system will allow you to look at ALL of the EMACS documentation files. The complete EMACS manual is available on-line by using this system.
- *help-with-tutorial* – (**C-h t**) The on-line tutorial takes you through all of the basic features of EMACS. This is a “learn-by-doing” style of tutorial. It is very easy to start and stop at any point in the lesson.
- *apropos* – (**C-h a**) *Apropos* is a more sophisticated way of getting help. A possible question one might have would be: “What are the commands for working with files?” To answer this, use the command (**C-h a file ↵**). This will list out all of the commands that contain the word *file*.

Copies of the GNU-EMACS manual are available from the RCF, however this manual can be accessed on-line via the *info* system as described above.