

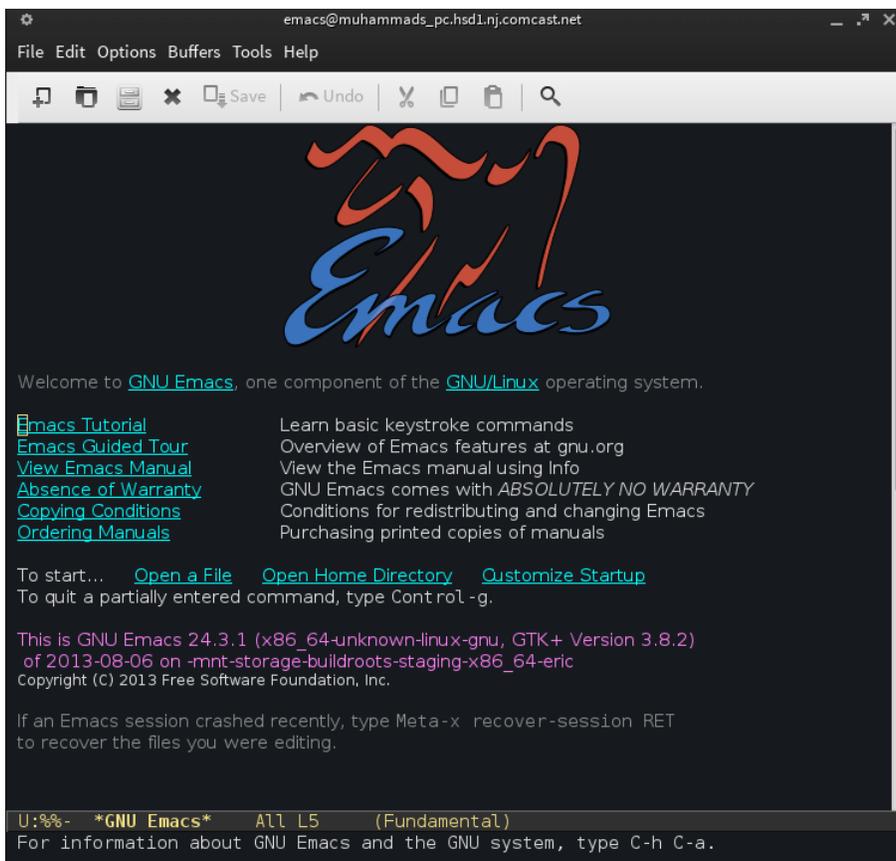
CS 3110 Guide to Emacs with Tuareg Mode

Introduction and Installation

Emacs is an extensible (customizable) text editor that has been referred to as “an operating system in itself” due to the multitude of features and power it provides the programmer. We shall be using in particular *GNU Emacs*, the most popular implementation of an Emacs editor.

- Windows users can visit <http://www.gnu.org/software/emacs/#Obtaining> to download the most recent stable release
- Mac users can download GNU Emacs from <http://emacsformacosx.com>, or choose to use *Aquamacs* (<http://aquamacs.org/>), an Emacs implementation tailored toward the Mac OS desktop environment
- Linux users can download from the same link as Windows users, but will most likely find the same edition in their respective package repositories.

If all went well, open the application, which should yield a window like this (colors may vary):



Basic Emacs Keystrokes

With Emacs now open, being a text editor and all, you essentially can just begin typing as you would to make your text appear. Just open a file, and start typing away - make sure to save periodically, though.

You may notice, however, that Ctrl+O isn't bringing up a dialog box for you to select your file. Emacs was around long before the conventions you are used to (Ctrl+C for "copy", Ctrl+X for "cut", Ctrl+V for "paste", etc.) came about, and so it has its own default keystrokes. Although you may initially find these keystrokes uncomfortable, you will soon realize that they are a lot more efficient and require less movement of your hands across the keyboard than their common counterparts.

In any Emacs documentation, you will find keystrokes to be in a format like the following: 'C - x C - f' or 'M - x doctor [RET]'. The 'C' refers to the **Control** key, and the 'M' refers to the **Alt** key ('M' stands for META, since early keyboards had this key instead of an 'Alt'). A set of characters joined by hyphens are meant to be pressed simultaneously, while a space indicates a release of all keys held down. Thus, the first command is executed first by pressing Ctrl+x, releasing these keys, and then pressing Ctrl+f. Likewise in the second example, you would press Alt+x, release the keys, type 'doctor' and lastly press the Return key.

It will be worth it to know at least the following keystrokes:

File Manipulation

C - x C - f	Visit file in current buffer: Equivalent of opening a file. Press Return to execute command after typing file path.
C - x C - s	Save current buffer
C - x s	Save all open files
C - x C - w	Equivalent of "Save As" (for current buffer only)
C - x k	<u>Kill current buffer:</u> Equivalent of closing file.

Text Manipulation

C – s	Search (forward) for text. Subsequent C – s will search for the next occurrence.
C – r	Search (backward) for text. Subsequent C – r will search for the previous occurrence.
M – % ... [RET] ...	Search-and-replace. First type search text, press Return, and type replacement text
C – k	K ill text from current position to end of line; saves to kill ring. Equivalent of “Cut”
C – y	Y ank most recently killed text at current position. Equivalent of “Paste”
C – [SPC]	Set MARK and begin defining region (highlighting text)
C – w	Kill entire region

File/Buffer Navigation

C – n	Move to n ext line
C – p	Move to p revious line
C – f	Move f orward one character
C – b	Move b ackward one character
C – a	Move to beginning of current line
C – e	Move to end of current line
C – v	Scroll down (approx.) one page
M – v	Scroll up (approx.) one page
M – <	Navigate to beginning of buffer. Sets MARK to be previous location
M – >	Navigate to end of buffer. Sets MARK to be previous location
C – x C – [SPC]	Navigate to MARK

Multi-Window Support

<code>C - x 0</code>	Close (but don't kill) current window
<code>C - x 1</code>	Close all but the current window
<code>C - x 2</code>	Split current window in half horizontally
<code>C - x 3</code>	Split current window in half vertically
<code>C - x o</code>	Switch to next window
<code>C - x b</code>	Re-open buffer in current frame (type the name yourself)
<code>C - x C - b</code>	List all open buffers. Navigate to desired and press Return to open in current frame

Miscellaneous

<code>C - g</code>	If in the process of typing command, abort it
<code>C - u < number > < command ></code>	Provide numerical argument to another command e.g. ' <code>C - u 3 C - n</code> ' moves 3 lines down
<code>C - x C - c</code>	Exit Emacs

Extensibility

The true extensibility of Emacs is possible through code written in a language called Emacs-Lisp (ELisp for short). The most important ELisp file is the `.emacs` file, which is loaded as soon as you open Emacs. On UNIX-based operating systems, the default location is `/home/ < you > /.emacs`. If this file is not found, Emacs will look for `/home/ < you > /.emacs.d/init.el`.

Tuareg Mode

Tuareg mode is an editing mode for Emacs that provides syntax highlighting, appropriate auto-indentation, and additional keystrokes when editing OCaml files.

We can use the `.emacs` file to load Tuareg mode automatically when Emacs starts up. First, to download the necessary files, visit the following URL:

<http://tuareg.forge.ocamlcore.org/>

Certain package repositories for Linux distributions already contain files for Tuareg mode, so it may be easier to download straight from there.

Once you have placed the files somewhere on your computer, add the following lines to your `.emacs` file:

```
(add-to-list 'load-path "/your/path/to/files/tuareg")
(autoload 'tuareg-mode "tuareg" "Major mode for editing Caml code" t)
(autoload 'ocamldebug "ocamldebug" "Run the Caml debugger" t)
(show-paren-mode 1)
(add-hook 'tuareg-mode-hook 'tuareg-imenu-set-imenu)
(setq auto-mode-alist (append '(("\\.ml[iy]p"?$" . tuareg-mode ("\\.topml$" . tuareg-mode))
auto-mode-alist))
```

Now, if you visit any OCaml file (.ml, .mli, etc) Emacs should automatically load Tuareg mode, and you should be good to go.

One huge benefit to Tuareg mode allows you to open up a toplevel in Emacs itself. To do so, simply execute

```
M - x tuareg - run - ocaml
```

and the toplevel should open up. Now, you can easily switch between your OCaml file and the toplevel to run a quick test. Furthermore, with a running toplevel, navigating to the beginning of an OCaml expression and executing `C - c C - e` will automatically cause that expression to be evaluated in the toplevel.

Further Reading

To learn more keystrokes not covered here, or more about Emacs in general, read the Emacs tutorial which you can visit [in Emacs itself](#) by executing the command

```
C - h t
```

For video tutorials, check out EmacsMovies.