Text Editing with Emacs- a really basic primer

Written by Dave Pawlowski

Introduction

Emacs is a free, portable, and extensible text editor that you will find on pretty much every Linux based system in existence. It works on a variety of architectures and other operating systems, and as it is quite portable, no matter what system you are on, you will find that the behavior of emacs is always the same. Emacs is extremely popular with programmers. If you use a common language, Emacs probably provides a mode that makes it especially easy to edit code in that language, which provides context sensitive indentation and layout, colors, and even the ability to run programming sessions, compile programs, debug your code, and interact directly with the language interpreter inside the Emacs editor itself.

A fantastic description of emacs resides at the GNU Emacs website. Additionally, I have prepared a more extensive emacs introduction that can be found at http://chuck.emich.edu/dpawlows/notes/emacs.pdf

Starting emacs

The following command entered on the Linux command line will start emacs and open a file (or create one if it doesn't exist)

```
% emacs temp.txt
```

Once you start an emacs session you can no longer interact with the command line. If you want to, but don't want to kill your emacs session completely, you can type:

```
% C-z
```

This will put the emacs session in the background so that you can interact with the command line. To get the emacs session back, type

```
% fg
```

Emacs commands you absolutely need to know

So you should have a file open. Type some stuff. Once you do that, you'll actually want to save your work.

Saving

> **C-x C-s**

will accomplish this. Note that command is two separate key strokes. You type control-x then let go and then type control-s in succession.

If you look at the bottom of the emacs window, emacs will tell you that you just saved a file.

Undo

Obviously, there will be a time when you need to undo a keystroke. This is accomplished using the

> **C-x u**

key binding.

Search

> **C-s**

will search forward in the document for a character string that you supply while

> **C-r**

will search backward.

Exiting Emacs

Finally, actually exiting the emacs process is accomplished by

> **C-x C-c**

Emacs is smart, so it will make sure you saved all the buffers that you were working on, and if you haven't it will ask you if you would like to.