Editing Documents

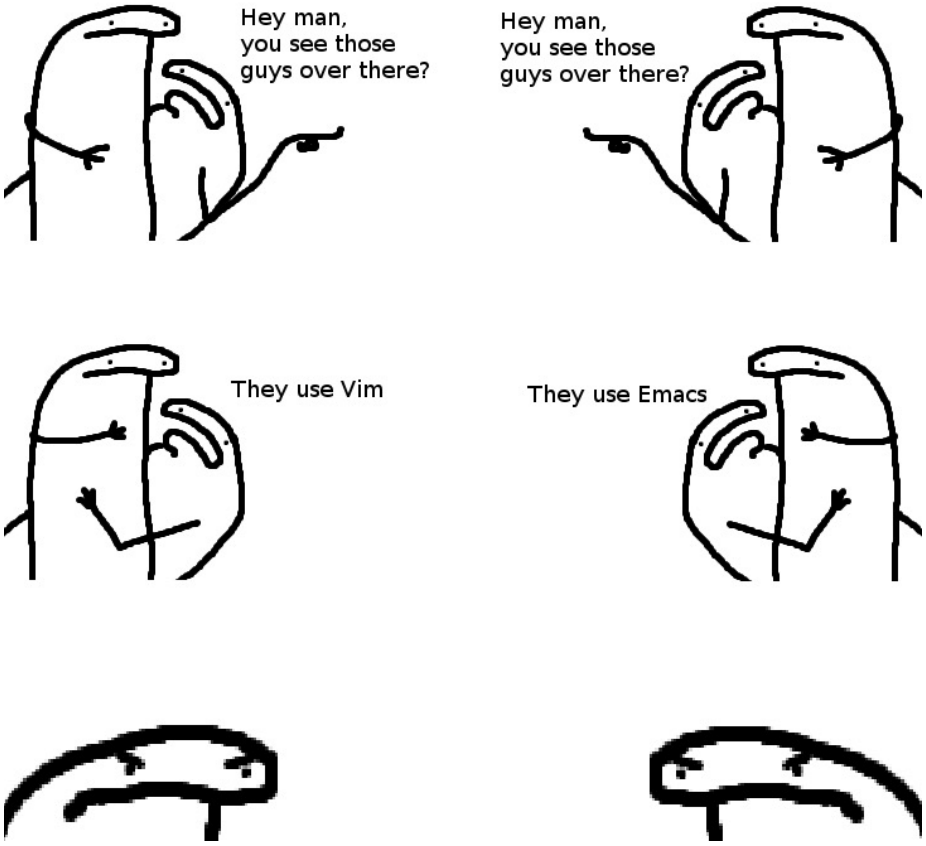

Figure 1: vim and emacs

Like most Linux nerds, I have gravitated to one of two major text editors: emacs and vim. The editor that eventually stole the show was vim. Since I started my Linux career, I have always used Atom which claimed to be a "Hackable" editor. It gave the impression that it would be easy to start writing plugins and making it super customizable. I ended up in a dead-end because I slowly realized this was an editor which was aimed for web developers and I was no such person. I had dabbled in HTML and some CSS but very surface level things. This made for a very steep learning curve and something I was not eagar to dive into this rabbit hole.

While using Atom, I grew to love the plugins I had acquired. I had been using nano *shudders* for some text editing while making scripts for my jailbroken iPhone, so I was not unfamiliar with the concept of terminal "IDEs". I decided

to get started with something better than nano and I had heard of vim through various forums and friends. After giving it a shot, I sort of liked it but I was super comfortable with visually selecting text with a mouse. I used it more and more along side Atom and grew tired of the slow responsiveness of Atom. It felt slow and was quickly losing it's appeal. After getting more familiar with vim, it quickly came apparent that this was a much better approach. It was fast, responsive and best of all; hackable.

Hacking

Back to those plugins. I adored them and it was the hardest thing to let go. For example, I had a compile/make run program for C/C++ files that was one button click magic. I once got curious on how this was written and realized that it merely called g++/clang++ (DUH!). I looked into how a key press can trigger an event and it was surprisingly simple. I then made it more specific (C/C++) files and presto!, the thing did what my make run plugin did in an entire project in one line. One. line. If you're curious, this is the final product:

```
autocmd FileType cpp nnoremap <buffer> <C-c>\

:!clear && clang++-6.0 -std=c++17 % -g -o %:r.out && ./%:r.out <Enter>
```

Note: "\" means line break

Here, the % stands for the filename of the current file being edited. With this, we are able to then call external shell commands with this variable. %:r means we want the filename without an extension. This emulates the following command:

```
clang++-6.0 -std=c++17 main.cpp -g -o main.out && ./main.out
```

This right here is what ultimately drove me to use vim. Anyone who is willing to read up on the documentation can make it function the way they want it to. I have always had this notion that "oh well, the editor is too complicated to configure, looks like I will just live without". Turns out, this is false.