Emacs and Advanced Tools in Unix

Drake M. Petersen and Brian Mesembe

October 10, 2017

Overview

The purpose of this course is to allow students to build and experience an environment that increases workflow and is tailored to the needs of the individual. Topics and material that will be covered in this course will teach students how to use and adapt to innovative tools that computer scientists use daily. These tools include the Unix command-line, Emacs and Vim text-editors, Git version control system, LaTeX typesetting system, and Shell scripting. Not only will the students learn how to use these tools, but practice using them in meaningful projects and assignments that will carry with them past the course and in their day to day lives as computer scientists.

The Course in Detail

The Primary focus of this course will be to use the GNU Emacs text editor to complete various tasks that would normally require multiple environments and work-spaces. Emacs is a powerful tool that goes beyond simple text editing for various languages. In this course students will learn to use Emacs as a word processor, interactive presentation display manager, excel-like spreadsheet manager, organization tool, extended Git version controller, and programming environment for all programming languages. In learning to use them students will also learn other Unix tools such as Git and shell scripting.

Emacs as a Workspace

Emacs has built-in tools that make it easy to use at a note taking enviornment, organized agenda calendar and todo list, and can be used to make spread sheets all in one file. Learning to use Emacs as a replacement for Office programs like Word and Excel is preferable because it has the added benefit of working as an agenda tool that increases workflow.

Every language has an IDE and each one works differently; short-cuts, auto-completion, etc., but in Emacs any project of any language can be developed in one editor with little to no configurations eneded. Students will learn to add new languages to their environment and be able to keep up workflow because Emacs will operate the same way they expect.

Unix Tools

Students will have an opportunity to learn and use Git, Vim, and Shell scripting to complete assignments. These tools will be taught in collaboration with the Emacs content so that they can 1) see the tools in action and 2) begin to build a customized work environment that is tailored to their needs.

Git will be used to version control their configuration. In learning this, students will have the knowledge to version control with git for any project in the future. Version control is a vital tool to use in the real world, especially when working in a team setting.

Other Tools

LaTeX is widely used in the STEM fields for papers and documents. Our intention with introducing LaTeX is for students to understand its fundamentals. From the basics, they will be able to use LaTeX with Emacs Org-mode, find and use LaTeX packages for various tasks, and make edits directly to .tex files that may not be exported correctly. Conveniently LaTeX exports seamlessly to HTML from Org-mode.

Elisp and Lisp are going to be programming languages that we will introduce but not discuss in great detail. Elisp is the primary language used by Emacs, and the purpose of introducing it so that students can further customize their Emacs environment. When discussing Elisp, Lisp must be introduced because Elisp is an extension of Lisp. Understanding how Lisp is read and written will shed light on why and how to use Elsip.

Assignments

Assignment Purposes

The students will have three short writing assignments that will be due in the second half of the semester. Two of the assignments will be about how they discovered and use a package in Emacs to help them in daily use. The reason we want to have this assignment is so that students can get comfortable writing about technology and become comfortable with sharing these findings. One of the two writing assignments will be presented in class. The final project is a small research assignment, that serves the same purpose, but is marginally more technical, where students explore something on their own. Many software developers have blogs and share many interesting things such as findings, tutorials, etc. These assignments are more about getting the students involved and comfortable with the community in which they take part.

Presentations

Toward the end of the course, students will be required to present a package that they have used in their workflow that they have enjoyed. In their presentation they will be require to used the 'demo-it' package that is used to develop interactive presentations. The same file they use to write their descriptions, code, and spreadsheets will be automatically turned into a slide show, where they will learn to add interactive shells and buffers. Learning to create a presentation in this manner makes it really easy, fast, and keeps everything organized.

Final Project

The final project will be a small research assignment of the students choosing. The assignment can be anything from exploring the complexity of an algorithm to comparing languages side by side. Minimum requirements for this project is that it should two pages long, customized latex export via org-mode, one code block, and one spreadsheet with auto-generated cells. This project will give students a chance to write a paper about something

they explored on their own, many computer scientist have blogs about personal projects and tutorials.

Conclusion

This course is designed to teach students how to develop and build a work environment that is personalized and familiar. All craftsmen from chefs to carpenters have dependable tools that get the job done well, as programmers our tools need to be the same way. Emacs is an amazing environment for everything that a computer scientists creates. Not only are the students going to learn how to use Emacs but other tools to create a complete tool belt for a variety of projects and needs. Our goal by the end of the course, is for every student to have the confidence in their tools and themselves to be great computer scientists.

Schedule Outline

Week	Content/Materials
0	Intro to Editor, Syllabus, Plagiarism
1	Vim basics, Using the cmd line
2	Intro to Emacs
3	Emacs customizing and packages, Setting up a Git repo
4	Emacs customizing cont. and package installing
5	Org-mode, LaTeX, Shell Scripting
6	More customization, Org-mode continued
7	Using Latex in an IDE, Midterm review
8	Midterm on Emacs commands, git, latex, cmd
9	Break
10	Presentation Day 1
11	Elisp/Lisp more in depth
12	Literate Programming and Reproducible Research
13	Presentation Day 2
14	Discussion Day: thoughts and feedback, other comments