Advanced - Editors

EECS 201 Fall 2024

Submission Instructions

Answer the bolded text that begins with a "**Q**". This assignment is an "online assignment" on Gradescope, where you fill out your answers directly.

1 Some Tools that Support Other Tools (6)

For this part, you may use Vim or Emacs, whichever you are more comfortable with.
Integrated Development Environments (IDEs), such as Visual Studio, Xcode, or Eclipse provide many features beyond simple text editing.[1] Some of the most useful features rely on an understanding of C/C++ code, allowing you to jump from a function's use to its definition, inspect the types of complex structures, and assist authoring by suggesting possible completions.
Vim and Emacs both support rudimentary autocompletion out-of-the-box. Try editing an existing file and start typing a function you've used before, then hit `ctrl-n` and `ctrl-p` in Vim or `M-/` in Emacs. This autocompletion has no understanding of C, it simply does string matching with any other strings anywhere else in the open files.[2]
There are two utilities, ctags/etags (Vim/Emacs respectively) and cscope that can be integrated with your editor to provide some contextually aware editing assistance. ctags/etags works pretty much out of the box. Somewhat annoyingly, cscope requires you to generate a list of files for it to process. You may find that the `find` utility, and its filetype filters in particular, are very helpful for generating this file list.
Set up ctags/etags and cscope for any project of your choosing, and integrate them with your editor of choice.

1. **Q:** What did you have to do to generate tags for ctags/etags? How often do you need to generate tags (after what changes do you need to regenerate tags)? What did you have to do to integrate ctags/etags with your editor so that you could navigate ctags from within your chosen editor (you might not actually need to do anything, and note if you don't need to do anything)?

2. **Q:** How did you generate a file list for cscope? How often do you need to run cscope? What did you have to do to integrate cscope with your editor so that you could utilize cscope functionality from within your chosen editor (you might not actually need to do anything, and note if you don't need to do anything)?

3. **Q:** Explain the difference between ctags/etags and cscope Is one strictly better than the other?

As a side note, nowadays we have the Language Server Protocol (LSP) which enables the use of a local server to provide services like code completion independent of text editors themselves, allowing text editors that support LSP to hook into language servers for features. In a future update to this assignment we'll look at those instead of the classic ctags and cscope. I just wanted to give it a mention here in case you were interested in the direction where text editors have been going.

[1] Interestingly, "simple text editing" is often what IDEs are worst at out-of-the-box. Hence part 3 of this assignment.
[2] Vim also understands how to parse C `#include` directives and Python `import` statements, and will search those files as well.

2 Extending Editors (2)

For this part, you may use any editor you want.
ctags/etags and cscope are programs that run independently of your text editor. You can also extend your editor with plugins. For Vim, plugins are written in the custom language Vimscript; Emacs itself is a Lisp interpreter, and plugins are written in Emacs Lisp. Sublime is written Python, as are its plugins. Atom in Javascript, et cetera.
While writing a plugin is a great way to learn a lot about the internals of your editor, it can be a steep learning curve. Fortunately, the Internet has an enormous library of really interesting and powerful plugins. As an example, there are plugins that solve the annoyance from part 1, that will automatically regenerate tags whenever needed.
Pick (at least) one plugin(s) to install in your editor.[3] You may find it is easier to first install a plugin manager and then install plugins. **If you installed a plugin for part 3, you may not count it for this part**.

1. **Q:** What editor are you using, What is this plugin's name, what does this plugin do and why did you install this plugin?

3 Everything Old is New Again (2)

While advanced IDEs and new text editors come out remarkably often, people who have learned how to navigate and manipulate text using Vim or Emacs are generally not willing to give that up. For this reason, there are plugins (or in some cases it's built-in) to provide Vim-like or Emacs-like editing modes for nearly every editor.
Pick your favorite "advanced" GUI editor (suggestions: Visual Studio, Xcode, Eclipse, Sublime Text, Atom, or Visual Studio Code – gVim doesn't count) and set it up to edit text in Vim or Emacs mode. You may need to find and install a plugin first.

1. **Q:** Explain how to set up and use Vim/Emacs mode for your chosen editor

[3] How to find them, choose? Try Googling "best plugins for (editor)", pick whatever seems cool to you.