

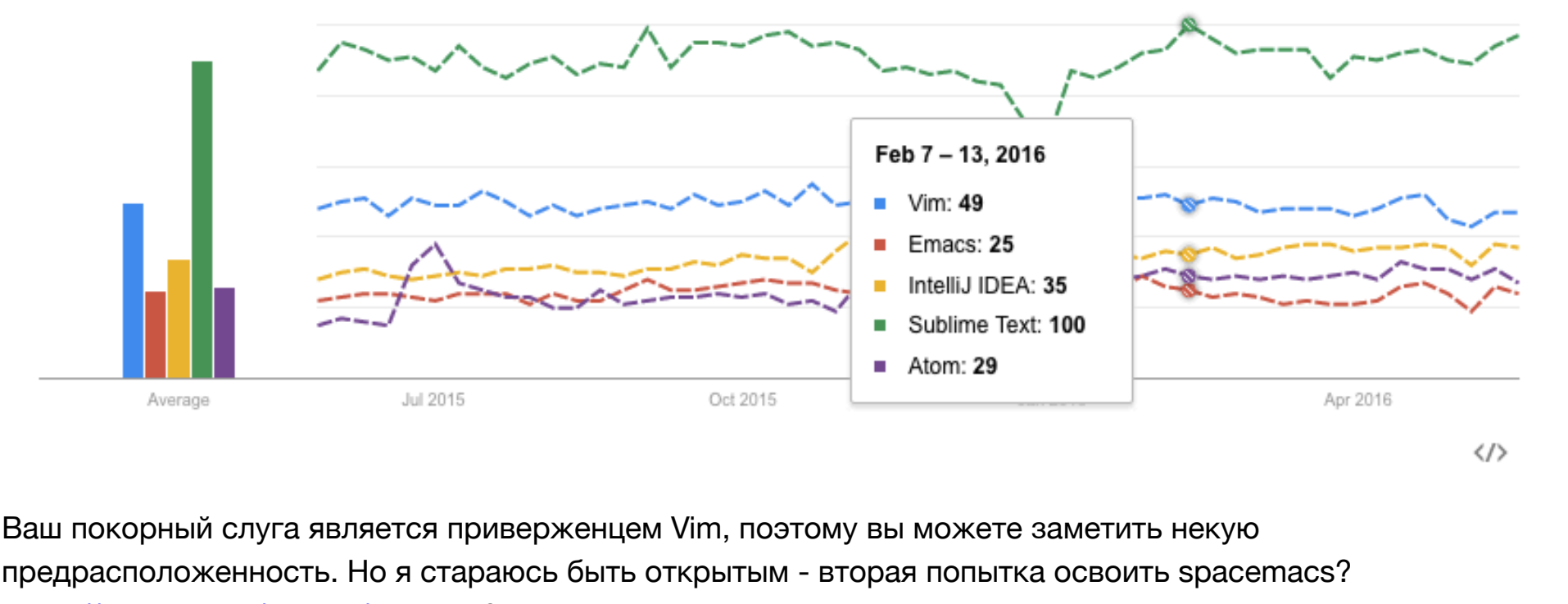
Введение

Что делает нас продуктивными при работе с кодом в редакторе? Каждый день появляется какой-то новый сияющий плагин, который позволяет нам сделать какую то операцию нажав лишь пару клавиш. Нет, я не против плагинов. Черт, у меня самого их по меньшей мере 40 (<https://git.io/vrg64>).

Но какой выхлоп получается на выходе? Совершенно точно, что **некоторые фишки дают больший профит нежели другие**. И это могут быть не только плагины, но и фишки, встроенные в редактор. Вы просто не знаете об их существовании. Здесь я постарался собрать те, которые по моему мнению **являются самыми важными** и не требуют чтения кучи документации и кучу времени на освоение.

Я не проводил никаких исследований, так что, боюсь, мне нечем подкрепить пункты ниже по тексту. Если вы думаете, что я упустил что-то, оставляйте свои комментарии.

Помимо этого, я дополнительно провел сравнение 5 наиболее распространенных редакторов. Я надеюсь, вам, как и мне, будет интересно узнать что происходит с другими редакторами.



Ваш покорный слуга является приверженцем Vim, поэтому вы можете заметить некую предрасположенность. Но я стараюсь быть открытым - вторая попытка освоить spacemacs? <https://github.com/google/xi-editor>? Кто знает..

1. Переключение между двумя последними файлами

Vim	<code>nnoremap <leader><leader> <C-^></code>
Emacs	<code>(global-set-key (kbd "M-o") 'mode-line-other-buffer)</code> https://www.emacswiki.org/emacs/SwitchingBuffers#toc5 http://emacsredux.com/blog/2013/04/28/switch-to-previous-buffer/
Idea	Ctrl + Tab / Ctrl + Shift + Tab http://blog.jetbrains.com/webide/2013/02/navigating-between-files-in-the-ide-best-practices/
Sublime Text	Ctrl + Tab / Ctrl + Shift + Tab
Atom	Ctrl + Tab / Ctrl + Shift + Tab

2. Запуск тестов не выходя из редактора

Vim	https://github.com/janko-m/vim-test
Emacs	нет единого интерфейса, но есть пакеты для разных языков (clojure - https://github.com/clojure-emacs/cider , ruby - https://github.com/pezra/rspec-mode , erlang - https://github.com/tjarvstrand/edts , python - https://github.com/ionrock/pytest-el , golang - https://github.com/nlamirault/gotest.el)
Idea	посредством UI плюс Ctrl + R для повторного запуска (для некоторых языков понадобится настроить Idea, например - golang - https://github.com/go-lang-plugin-org/go-lang-idea-plugin/wiki/Documentation) https://www.jetbrains.com/help/idea/2016.1/rerunning-tests.html)
Sublime Text	нет единого интерфейса, но есть пакеты для разных языков (ruby - https://github.com/maltize/sublime-text-2-ruby-tests , python - https://damnwidget.github.io/anaconda/tests_runner/ , scala - https://github.com/jarhart/SublimeSBT)
Atom	нет единого интерфейса, но есть пакеты для разных языков (ruby - https://github.com/moxley/atom-ruby-test , http://brendankemp.com/essays/atom-is-ready-to-be-your-editor/ , elixir - https://github.com/indiejames/atom-iex , (many) - https://atom.io/packages/script)

3. Быстрое переключение между терминалом (или эмулятором) и редактором

Console Vim and all other terminal text editors	Ctrl + z & fg Если вам лень набирать fg, то есть небольшой хак: zsh: https://github.com/robbyrussell/oh-my-zsh/blob/master/plugins/fancy-ctrl-z/fancy-ctrl-z.plugin.zsh fish: <pre>bind \ck 'if test -z (commandline) ; fg %; else ; clear; commandline ""; end'</pre>
MacVim (vim-gnome, т.п.)	нет способа
Neovim	как переход на обычное окно
Emacs	как переход на обычное окно
Idea	как переход на обычное окно
Sublime Text	как переход на обычное окно https://packagecontrol.io/packages/Terminal
Atom	как переход на обычное окно https://atom.io/packages/term3

4. Переход к определению

Vim	Ctrl +] / Ctrl - T используя ctags
Emacs	M-. / M-* используя ctags https://www.emacswiki.org/emacs/EmacsTags или https://github.com/jacktasia/dumb-jump используя ag
Idea	Ctrl + B или Ctrl + click / Ctrl + Tab
Sublime Text	Ctrl + click / Ctrl + right click https://gist.github.com/kendellfab/6135193#gistcomment-1768760
Atom	Ctrl + Alt + down / Ctrl + Alt + up https://github.com/atom/symbols-view

Генерирование тэгов посредством git хуков: <http://tbaggery.com/2011/08/08/effortless-ctags-with-git.html>

Секция для продвинутых

5) REPL

Если REPL запущена рядом (или внутри) редактора - отлично! Можно eval'ить куски кода, смотреть документацию (<http://tonsky.me/blog/interactive-development/>).

Vim	отсутствует
Neovim	есть для многих языков https://github.com/kassio/neoterm
Emacs	есть для многих языков (ruby - https://github.com/dgutov/robe , scala - https://github.com/ensime/ensime-emacs , clojure - https://github.com/clojure-emacs/cider)
Idea	есть для многих языков (clojure - https://cursive-ide.com/)
Sublime Text	есть для многих языков https://packagecontrol.io/packages/SublimeREPL
Atom	есть лишь для некоторых языков (clojure - https://atom.io/packages/proto-repl , haskell - https://atom.io/packages/ide-haskell-repl)

Вот такой получился список:

- 1) Переключение между двумя последними файлами
- 2) Запуск тестов не выходя из редактора
- 3) Быстрое переключение между терминалом (или эмулятором) и редактором
- 4) Переход к определению
- 5) REPL

Пункты 3 и 5 могут быть решены посредством терминальных мультиплексоров (на русском некоторые вещи звучат супер странно). Например, моего любимого <https://tmux.github.io/>.

Бонус

- 1) Не повторяйтесь ("." в Vim, макросы [atom - <https://atom.io/packages/atom-keyboard-macros>])
- 2) Опирируйте сущностями: блоками, функциями, т.п. (motions и текстовые объекты <https://github.com/kana/vim-textobj-user/wiki> в Vim)
- 3) Быстро перемещайтесь по коду с помощью AceJump подобных плагинов (vim - <https://github.com/easymotion/vim-easymotion>, idea - <https://github.com/johnlindquist/AceJump>, emacs - <https://github.com/abo-abo/avy>, sublime text - <https://packagecontrol.io/packages/EasyMotion>, atom - <https://atom.io/packages/jumpy>). Мое текущее использование vim-easymotion сводится лишь к одной команде <https://git.io/vrzAE>, которая позволяет мне прыгать в любое место в текущем файле. Просто и эффективно!
- 4) Используйте несколько курсоров (vim - <https://github.com/terryma/vim-multiple-cursors>, idea - F6 и остальные, emacs - <https://github.com/magnars/multiple-cursors.el>, sublime text - Ctrl + D, atom - Ctrl + D)

Заключение

Как вы видите, каждый из перечисленных редакторов неплохо справляется со своими задачами. Некоторые чуть лучше, другие - чуть хуже.

Еще один вывод: Vim больше всего подходит для полиглот разработчика (<http://searchsoftwarequality.techtarget.com/definition/polyglot-programming>), поскольку предоставляет плагины и один интерфейс для множества языков. Емакс предлагает больше мощи, но его придется донстроить, чтобы маппинги были одинаковыми. Может это хорошая идея для Spacemacs, как думаете?

Полезные ссылки:

- <https://sheerun.net/2014/03/21/how-to-boost-your-vim-productivity/>
- <https://danielmiessler.com/blog/enhancements-to-shell-and-vim-productivity/>

Я что-то упустил? Оставляйте комментарии на <http://homeonrails.com/> или напишите мне email.