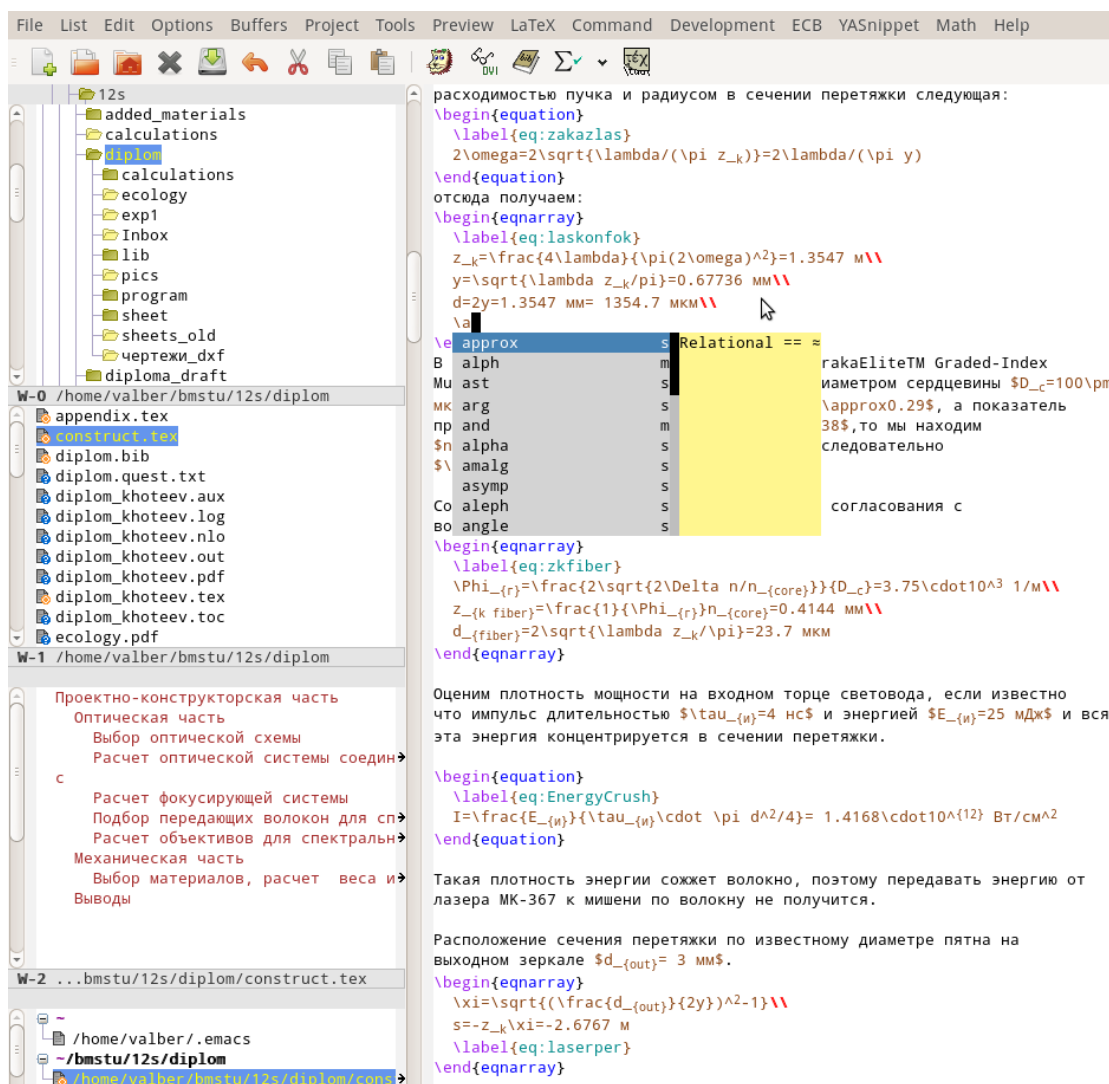


Мой конфиг для Emacs

Valber :



Вроде как год прошел с начала использования Emacs 23(24) для себя и он не перестает меня удивлять. Сразу оговорюсь я не гуру, да я прочитал уроки по elisp благо они на русском валяются в интернете, но я ничего не писал с нуля. И мой конфиг представляет из себя одеяло из лоскутков, но оно работает .

К сожалению статья вышла позже статьи про org-mode , потому как [org-mode](#) это тоже своя вселенная и создавая новые статьи, я нет нет, да открою для себя что нибудь)

Emacs for Python

Замечательное творение Gabriel Lanago , действительно спасло меня в первые дни уже автоматически включает прекрасное автодополнение и протаскивает с собой необходимые пакеты

вроде as-compilite, настройки автодополнения настолько прекрасны и общи что работали у меня и в С и в Emacs.

Официальный репозиторий

Зависимости:

- emacs 23
- pyflakes
- pymacs (optional but extremely recommended)

```
;; Python - мод , активирующий автодополнение и прочие прелести.
(load-file "~/.emacs.d/emacs-for-python/epy-init.el")
(add-to-list 'load-path "~/.emacs.d/emacs-for-python/") ;; tell where to load
the various files
(require 'epy-setup) ;; It will setup other loads, it is required!
(require 'epy-python) ;; If you want the python facilities [optional]
(require 'epy-completion) ;; If you want the autocompletion settings
[optional]
(require 'epy-editing) ;; For configurations related to editing [optional]
(require 'epy-bindings) ;; For my suggested keybindings [optional]
(epy-setup-checker "pyflakes %f")
(epy-django-snippets)
;; (epy-setup-ipython)
(require 'highlight-indentation)
(add-hook 'python-mode-hook 'highlight-indentation)
```

ECB

Emacs Code Browser — добавляет всякие окошечки , которые что приятно помимо настраиваемости работают и без иксов в псевдографическом режиме. Не знаю как в других дистрах а в Ubuntu, ECB намертво привязан к [CEDET](#) — это такой проект в основном для C,C++ (говорят ещё и Java), в который входят по мимо ECB , синтаксические анализаторы кода, менеджер проектов, построитель UML диаграмм классов, в общем для меня его использование это сильное колдунство.

У меня почему-то сразу не срабатывал анализатор кода для Python , пришлось повесить следующую заглушку (спасибо ЛОР).

```
;;Анализ кода с помощью CEDET и указание в окне ECB используемых
;;классов и переменных
(semantic-mode -1) ;; Отключаем для проверки
(require 'ecb)
(add-hook 'ecb-activate-hook (lambda () (semantic-mode t))) ;; Включаем в
хуке
```

Комбинации:

- **M-x ecb-activate** Превращает Emacs из блокнотика в многооконное нечто)
- **M-x ecb-deactivate** Отключает окна ECB
- **C-c . g d** Перейти в буфер с каталогами
- **C-c . g l** Перейти в буфер редактирования
- **C-c . g m** Перейти в буфер Методы и переменные которые содержит заголовки содержащихся в файле функций или в случае LaTeX секции и главы.
- **C-c . g s** Перейти в буфер в котором показаны файлы содержащиеся в текущем каталоге
- **C-c . g h** Перейти в буфер истории(здесь обычно скапливаются все открытые буферы)

AUCTeX

Пакет для работы с Tex, LaTeX, XeTeX и прочим, обладает массой полезных комбинаций, а так как автодополнение и снипеты(шаблонные конструкции) уже включены пакетом `emacs-for-python`, нам остается только наслаждаться. Хотя придется скачать auto-complete-auctex.el Уж очень хотелось чтобы при дополнении показывали изображения знаков и греческих символов.

```
;;Автодополнение для AUCTeX
(load-file "~/emacs.d/auto-complete-auctex/auto-complete-auctex.el")

(custom-set-variables
 ;; custom-set-variables was added by Custom.
 ;; If you edit it by hand, you could mess it up, so be careful.
 ;; Your init file should contain only one such instance.
 ;; If there is more than one, they won't work right.
 '(LaTeX-symbols-active-menuitem 1)
 '(TeX-bar-LaTeX-buttons (quote (new-file open-file dired kill-buffer save-
buffer undo cut copy paste [separator nil] latex next-error view bibtex
latex-symbols-experimental nil)))
 '(TeX-view-program-selection (quote (((output-dvi style-pstricks) "dvips and
gv") (output-dvi "Evince") (output-pdf "Evince") (output-html "Evince"))))
 '(ecb-auto-compatibility-check nil)
 '(ecb-options-version "2.40")
 '(ecb-source-path (list (list "~" "~")))
 '(ecb-tip-of-the-day nil)
 '(gdb-many-windows t)
 '(gdb-use-colon-colon-notation t)
 '(latex-run-command "latex"))
(custom-set-faces
 ;; custom-set-faces was added by Custom.
 ;; If you edit it by hand, you could mess it up, so be careful.
 ;; Your init file should contain only one such instance.
 ;; If there is more than one, they won't work right.
 )
```

Также советую покопаться в настройках AUCTeX, там вы можете активировать специальное меню Math со списком всех символов.

Комбинации:

Их конечно больше но все же

- **C-c C-t C-p** Сменить вывод с DVI на PDF
- **C-c C-c** Запуск компилятора(pdf latex например), также устанавливает текущий файл master файлом(спец метка в конце файла)
- **C-c C-e** Вставить конструкцию вроде формул или списков с возможностью автодополнения названия.

```
\begin{} \end{}
```

- **C-c C-s** Вставить один из элементов оглавления.
- **C-c C-f C-b** Вставить жирный шрифт
- **C-c C-f C-i** Вставить курсив
- **C-c Стрелка_вверх** Переключиться на мастер файл, мастер файл это исходный корневой файл вашего проекта который включает в себя все остальные подкаталоги

Сложная сборка DVI->PS->PDF

Стандартная сборка позволяет собрать либо DVI либо PDF , но вот когда мы используем нечто например PStricks нам нужен порядок компиляции по сложнее. Один из вариантов решений при сложной компиляции это создание Makefile и запуска его командой **M-x compile** или в терминале.

```
(setq my-tex-commands-extra (list
                             (list "Custom Compile"
                                     "latex -interaction=nonstopmode %s.tex && latex -interaction=nonstopmode
                                     %s.tex && dvips %s.dvi -o %s.ps && ps2pdf %s.ps %s.pdf"
                                     'TeX-run-command nil t)))
(require 'tex)
(setq TeX-command-list (append TeX-command-list my-tex-commands-extra))
```

После этого набрав в **C-c C-c** , введите в первый раз *Custom Compile*

Русификация

К сожалению по мимо руководства и справок в интернете, меню в Emacs на русифицировать(пока нет такой возможности). Зато можно сделать так чтобы не переключаться ,при введении комбинаций клавиш, на английский, а сразу вводить их на русском.

Для Emacs 23(в 24 входит по умолчанию). [Взято отсюда](#)

```
;;Функция ввода командных комбинаций без переключения раскладки M-x
;;reverse-input-method RET russian-computer RET
(defun reverse-input-method (input-method)
```

```
"Build the reverse mapping of single letters from INPUT-METHOD."
(interactive
 (list (read-input-method-name "Use input method (default current): ")
 (if (and input-method (symbolp input-method))
     (setq input-method (symbol-name input-method)))
 (let ((current current-input-method)
       (modifiers '(nil (control) (meta) (control meta))))
   (when input-method
     (activate-input-method input-method))
   (when (and current-input-method quail-keyboard-layout)
     (dolist (map (cdr (quail-map)))
       (let* ((to (car map))
              (from (quail-get-translation
                      (cadr map) (char-to-string to) 1)))
         (when (and (characterp from) (characterp to))
           (dolist (mod modifiers)
             (define-key local-function-key-map
              (vector (append mod (list from)))
              (vector (append mod (list to))))))))))
   (when input-method
     (activate-input-method current)))))
```

Словарь

Хотел бы тут добавить что изначально **ispell-buffer** начинает последовательную проверку каждого слова, что весьма неудобно, т.к. часто могут попадаться английские слова и прочее, поэтому я включая после написания **flyspell-buffer** он подсвечивает все слова с неправильной орфографией, а уже просмотреть варианты можно перейдя к слову и нажав **M \$** (Alt-Shift-4)

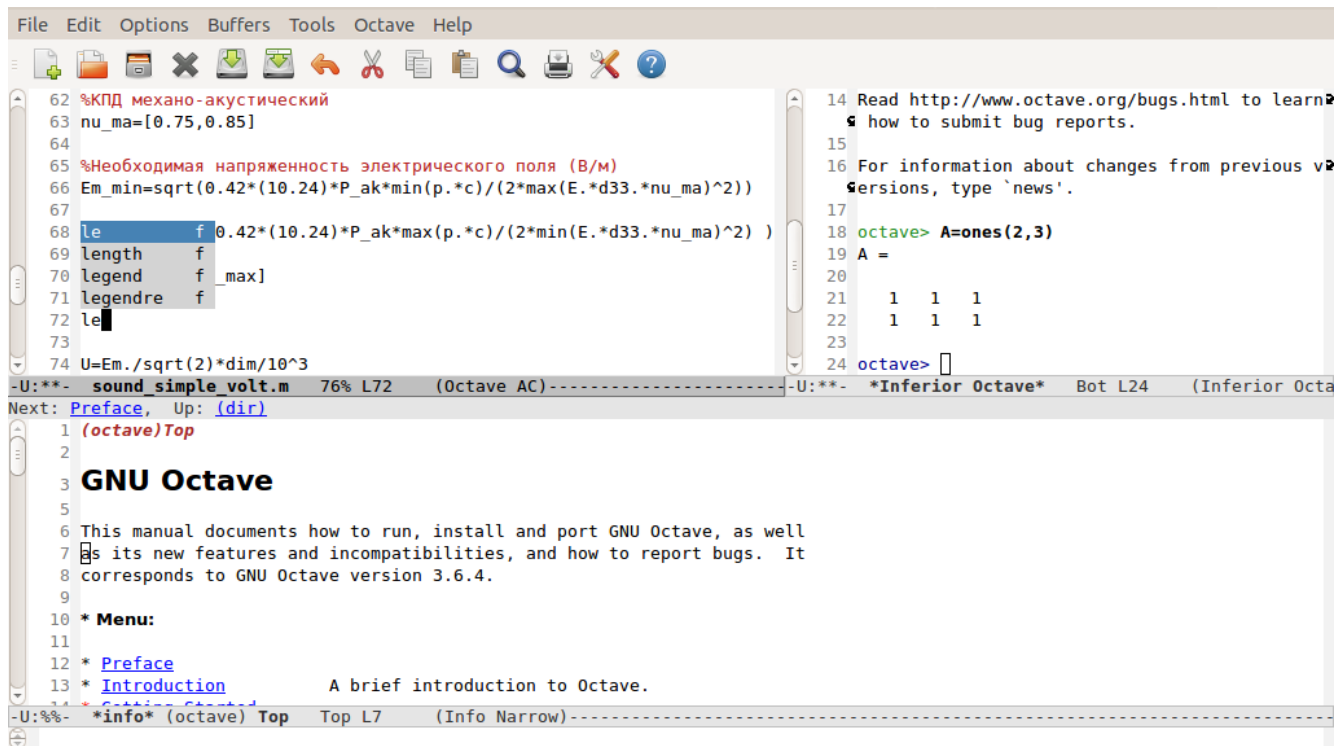
```
;; Провека орфографии
(setq flyspell-dictionary "russian")
;; Клавиши проверки орфографии
;; (global-set-key [f9] `ispell-buffer)
(global-set-key [f9] `flyspell-buffer)
(global-set-key [f8] `flyspell-mode)
```

Octave Emacs Support

Для этого придется скачать файл и подключить его

```
;; Автодополнение для Octave
(load-file "~/emacs.d/ac-oct ave.el")
```

Подробнее читай в обзорной статье к [Octave EOS](#)



Поддержка всяких интересных синтаксисов

Wikipedia

Файл ищется на самой википедии

```

;; Добавление под светки д ля .wiki файлов
(load-file "~/emacs.d/wikipedia-mode.el")
(add-t o-list 'aut o-mode-alist
'("\.wiki\\"" . wikipedia-mode))

```

AsciiDoc

```

;; Добавление под д ержи asciidoc формата используемого в д окументации
linuxcnc
(load-file "~/emacs.d/markup-faces/markup-faces.el")
(load-file "~/emacs.d/adoc-mode/adoc-mode.el")

```

G-code и HAL

G-code стырил в интернете... пока единственная реализация, но это надо пилить.

```

;; Подсветка G-кода
;; Набор сделай сам http://www.emacswiki.org/emacs/ModeTutorial
(define-generic-mode gcode-generic-mode
'(("(" . ")"))
(apply 'append
(mapcar #'(lambda (s) (list (upcase s) (downcase s) (capitalize s)))
'("sub" "endsub" "if" "do" "while" "endwhile" "call" "endif")

```

```

"sqrt" "return" "mod" "eq" "ne" "gt" "ge" "lt" "le" "and"
"or" "xor" "atan" "abs" "acos" "asin" "cos" "exp"
"fix" "fup" "round" "ln" "sin" "tan" "repeat"
"endrepeat"))))
'(("\\(#<_[A-Za-z0-9_]+>\\)" (1 font-lock-type-face))
  ("\\([NnGgMmFfSsTtOo]\\)" (1 font-lock-function-name-face))
  ("\\([XxYyZzAaBbCcUuVvWwIiJjKkPpQqRr]\\)" (1 font-lock-string-face))
  ("\\([\\-+]?[0-9]*\\.?[0-9]+\\)" (1 font-lock-constant-face))
  ("\\(#[0-9]+\\)" (1 font-lock-type-face))
  ("\\([0-9]+\\)" (1 font-lock-constant-face)))
'("\\.ngc\\'")
nil
"Generic mode for g-code files.")

```

Также подсветка .hal файлов. Файлы настроек и связи с железом в LinuxCNC, самописное.

```

;;Подсветка синтаксиса для HAL в LinuxCNC
;;GPLv2
(define-generic-mode hal-linuxcnc-generic-mode
  '("#".?\\n) ("/*"."*/"))
  ("or" "and")
  '(";" "\\(#<_[A-Za-z0-9_]+>\\)" (1 font-lock-type-face))
  '(";" "\\([NnGgMmFfSsTtOo]\\)" (1 font-lock-function-name-face))
  ("\\(net\\|loadrt\\|loadusr\\|addf\\|setp\\|linksp\\|newsig\\)" (1 font-
lock-function-name-face))
  ("\\
(axis\\|stepgen\\|parport\\|pwmgen\\|iocontrol\\|motion\\|classicladder\\)"
(1 font-lock-builtin-face))
  ("\\([ \\t]+--[A-Za-z0-9_]+\\)" (1 font-lock-string-face))
  ("\\(AXIS_\\|EMCMOT\\|TRAJ\\)\\(servo\\|base\\)-thread" (1 font-lock-
constant-face))
  ("\\(step\\|dir\\|enable\\)" (1 font-lock-type-face))
  ("\\([0-9]+\\)" (1 font-lock-constant-face))
  )
'("\\.hal\\'")
nil
"Generic mode for .hal file. LinuxCNC HAL")

```

Красивости

Подсветка синтаксиса

Ну можно поставить пакет emacs-goodies-el в него входит множество цветовых тем

```

;; Под грузка цветовой схемы
;;(color-theme-initialize)

```

```
;; Запустить понравившуюся тему gnome2
;; (color-theme-gnome2)
```

А можно поставить [solarized](#)

```
;; Цветовая тема Solarize
;; (load-file "~/.emacs.d/emacs-color-theme-solarized/color-theme-
solarized.el")
;; (require 'color-theme-solarized)
;; (color-theme-solarized-dark)
```

Minimap

Новомодные веяния от любителей SublimeText, инструкции приведены по [ссылке](#) .

```
;; Создавать Minimap как в sublime текст... ну мне не надо, так чтобы
;; модников заткнуть http://www.emacswiki.org/emacs/MiniMap
;; Use 'M-x minimap-create' in a buffer you're currently editing.
;; Use 'M-x minimap-kill' to kill the minimap.
;; Use 'M-x customize-group RET minimap RET' to adapt minimap to your needs.
(load-file "~/.emacs.d/minimap/minimap.el")
(require 'minimap)
```

Pony-Mode

Это собственно мод для работы с Django, [скачать можно отсюда](#) . За подробной информацией по [аналогам обращайтесь к wiki](#) проекта Django

```
;; Работа с django с помощью pony-mode
(add-to-list 'load-path "~/.emacs.d/pony-mode/src")
(require 'pony-mode)
```

Что позволяет, обновлять базу данных, запускать development-server, совершать миграцию, открывать браузер , останавливать сервер, переходит к настройкам проекта, делать дампы данных в json формате. во всем и не разберешь не с моим уровнем владения Django.

Eproject

у меня поставился с каким то пакетом

Eproject — это такая упрощенная версия [менеджера проектов](#) , собственно менеджера проектов есть и CEDET ... но это слишком сложно.

Eproject обладает простыми текстовыми настройками например заходим в пункт **Setup** .

И вводим название проекта и адрес корневой директории с ним.


```
Projects - Revert Save Quit
```

```
--
```

```
srctest1          : /home/valber/forgе/pmg/django/srctest1
slider            : /home/valber/forgе/pmg/PyQt/slider
```

Теперь зайдём **Setup** проекта. Вот так например выглядят все файлы проекта, которые откроются в emacs при открытии проекта.

```
Files - Revert Save Quit
```

```
- slider -
```

```
/home/valber/current_work/EOS.html
/home/valber/.emacs
/home/valber/bmstu/11s/below/ecomonitor_quest.tex
```

Теперь рассмотрим секцию **Tools**, часть из настроек создается по умолчанию, вы их можете только отредактировать.

```
Tools - Revert Save Quit
```

```
- slider -
```

```
Make (f9)          : make
Clean (C-f9)       : make clean
Run (f8)           : python main.py
Stop (C-f8)        : -e eproject-killtool
---               :
Configure          : ./configure
---               :
Explore Project    : nautilus --browser `pwd` &
XTerm In Project   : xterm &
```

После этого меню Project расширяется и дополняется всеми этими командами и сочетаниями клавиш, просто и удобно.

Послесловие

Ещё я использую **Graphviz** и **Gnuplot** мод, очень удобно, подсветка синтаксиса, генерация картинок и просмотр их в буфере, также просмотрщик документации **Info** файлов великолепен.

Сознательно не рассматривал здесь , проверку почты, jabber/irc клиент ,редактор видео, навигатор, психоаналитик и другие возможности Emacs .. ну все же мне нужен был редактор текста и кода а не надстройка над операционной системой...

Жду Ваших замечаний и предложений!!