Emacs Configuration

Nguyễn Đức Hiếu

Ngày 23 tháng 5 năm 2017

1 Information

The whole thing is driven by the following `.emacs`

```
;; init.el for this setup. Must use Emacs 24
(org-babel-load-file
(expand-file-name "init.org"
                  "/media/Logical_Drive/Emacs/Settings"))
```

2 General

General config for emacs with no major mode involved

2.1 Path configurations

```
(require 'package)
(add-to-list 'package-archives '("melpa" . "http://melpa.org/packages/"))

(package-initialize)

;;
(require 'server)
(unless (server-running-p)
  (server-start))
```

2.2 Appearence

```
;; Initialize Emacs full screen
```

```
(add-to-list 'initial-frame-alist '(fullscreen . maximized))

;; No startup messages on *scratch* buffer
(setq initial-scratch-message "")

;; Cursor type
(setq-default cursor-type 'bar)


;; Default font
(require 'unicode-fonts)
(unicode-fonts-setup)
(set-frame-font "DejaVu Sans Mono 10" nil t)


;; Set themes
(load-theme 'gruvbox t)
(set-face-attribute 'font-lock-comment-face nil :foreground "#27ae60")
(set-face-attribute 'default nil :foreground "#ffffff")
(set-face-attribute 'mode-line nil :background "#2f4f4f" :foreground "#ffffff")

;; Set background face for color string
(add-hook 'prog-mode-hook 'rainbow-mode)


;; Global font-lock mode
(setq global-font-lock-mode t)


;; Enable line number and column number
(setq column-number-mode t)

;; Disable tool bar, menu bar, and scroll bar
(tool-bar-mode -1)
(scroll-bar-mode -1)
(menu-bar-mode 1)

;; Paren mode
(show-paren-mode t)
(setq show-paren-delay 0)
```

2.3 Editing

```
;; Delete marked region when input
(delete-selection-mode 1)

;; Global mark ring
(setq global-mark-ring-max 50000)

;; "Yes or no"? Too much writing
(defalias 'yes-or-no-p 'y-or-n-p)


;; Auto close bracket insertion.
(electric-pair-mode 1)
(setq electric-pair-pairs '(
    (?\" . ?\")
    (?\( . ?\))
    (?\{ . ?\})
    ) )

(when (fboundp 'electric-indent-mode) (electric-indent-mode -1))


;; Set kill ring size
(setq global-mark-ring-max 50000)

;; Code completion
(require 'company)
(require 'company-files)
(add-hook 'after-init-hook 'global-company-mode)

(setq company-selection-wrap-around t
      company-tooltip-align-annotations t
      company-idle-delay 0.36
      company-minimum-prefix-length 2
      company-tooltip-limit 10)
```

```
;; Bound undo to C-z
(global-set-key (kbd "C-z") 'undo)

;; Expand region with M-m
(require 'expand-region)
(global-set-key (kbd "C-'") 'er/expand-region)

;; Multi-cursor
(require 'multiple-cursors)
(global-set-key (kbd "C-M") 'mc/edit-lines)
(global-set-key (kbd "C->") 'mc/mark-next-like-this)
(global-set-key (kbd "C-<") 'mc/mark-previous-like-this)
(global-set-key (kbd "C-c C-<") 'mc/mark-all-like-this)

;; Define function: fill character to 80
(defun fill-to-end (char)
  (interactive "cFill Character:")
  (save-excursion
    (end-of-line)
    (while (< (current-column) 80)
      (insert-char char))))
```

2.4 Miscelanous

```
;; Auto-revert mode
(global-auto-revert-mode 1)
(setq auto-revert-interval 0.5)
```

```
;; Backup stored in /tmp
(setq backup-directory-alist
      `((".*" . ,temporary-file-directory)))
(setq auto-save-file-name-transforms
      `((".*" ,temporary-file-directory t)))

;; Delete old backup
```

```
(message "Deleting old backup files...")
(let ((week (* 60 60 24 7))
      (current (float-time (current-time))))
  (dolist (file (directory-files temporary-file-directory t))
    (when (and (backup-file-name-p file)
       (> (- current (float-time (fifth (file-attributes file))))
  week))
      (message "%s" file)
      (delete-file file))))




;; Startup
(add-hook 'after-init-hook
  (lambda ()
  (find-file "/media/Logical_Drive/Emacs/Settings/init.org")))
```

2.5 Yasnippets

```
;; Enable Yasnippets
(require 'yasnippet)
(yas-global-mode 1)
```

2.6 Helm

```
(require 'helm)
(require 'helm-config)

;; The default "C-x c" is quite close to "C-x C-c", which quits Emacs.
;; Changed to "C-c h". Note: We must set "C-c h" globally, because we
;; cannot change `helm-command-prefix-key' once `helm-config' is loaded.
(global-set-key (kbd "C-c h") 'helm-command-prefix)
(global-unset-key (kbd "C-x c"))

(define-key helm-map (kbd "<tab>") 'helm-execute-persistent-action) ; rebind tab to r
```

```
(define-key helm-map (kbd "C-i") 'helm-execute-persistent-action)   ; make TAB work i
(define-key helm-map (kbd "C-z")  'helm-select-action)              ; list actions us

(when (executable-find "curl")
(setq helm-google-suggest-use-curl-p t))

(setq
helm-split-window-in-side-p           t ; open helm buffer inside current window, not
helm-move-to-line-cycle-in-source     t ; move to end or beginning of source when rea
helm-ff-search-library-in-sexp        t ; search for library in ‘require’ and ‘declar
helm-scroll-amount                    8 ; scroll 8 lines other window using M-<next>/
helm-ff-file-name-history-use-recentf t
helm-echo-input-in-header-line        t)


(setq helm-autoresize-max-height 0)
(setq helm-autoresize-min-height 30)
(helm-autoresize-mode 1)

(helm-mode 1)

;; Use helm for some common task
(global-set-key (kbd "C-x b") 'helm-buffers-list)
(global-set-key (kbd "M-x") 'helm-M-x)
(global-set-key (kbd "C-x C-f") 'helm-find-files)
(setq helm-M-x-fuzzy-match t)


;; Use "C-:" to switch to Helm interface during companying
(require 'helm-company)
(eval-after-load 'company
'(progn
(define-key company-mode-map (kbd "C-:") 'helm-company)
(define-key company-active-map (kbd "C-:") 'helm-company)))
```

3 Org-mode

```
;; Enable shift selection
(setq org-support-shift-select t)


;; fontify code in code blocks
(setq org-src-fontify-natively t)
(set-face-attribute 'org-block nil :foreground "#ffffff")
(set-face-attribute 'org-block-begin-line nil :foreground "#d5c4a1")
(set-face-attribute 'org-block-end-line nil :foreground "#d5c4a1")
```

4 Terminal

```
;; Keybinding for terminal
(global-set-key [f1] 'shell)

;; Use ubuntu font
(add-hook 'shell-mode-hook (lambda ()
   (setq buffer-face-mode-face '(:family "Ubuntu"))
  (buffer-face-mode)))
```

5 ESS for R programming

5.1 Setting up

```
(require 'ess-site)
(require 'ess-rutils)

(add-hook 'inferior-ess-mode-hook (lambda () (font-lock-mode 0)) t)

;; Indentation style
(setq ess-default-style 'RStudio)

;; Describe object
(setq ess-R-describe-object-at-point-commands
```

```
    '(("str(%s)")
     ("print(%s)")
     ("summary(%s, maxsum = 20)")))
```

5.2 Code completion

```
(setq ess-use-company 'script-only)
(setq ess-tab-complete-in-script t) ;Press <tab> inside functions for completions

;; Show quickhelp
(define-key company-active-map (kbd "C-;") 'company-show-doc-buffer)

;; Others
(setq company-selection-wrap-around t
company-tooltip-align-annotations t
company-idle-delay 0.36
company-minimum-prefix-length 2
company-tooltip-limit 10)

;; Eldoc mode for function arguments hints
(require 'ess-eldoc)
```

5.3 Functions and key bindind

```
;; Remap "<-" key to M-- instead of smart bind to "_"
(ess-toggle-underscore nil)
(define-key ess-mode-map (kbd "M--") 'ess-smart-S-assign)
(define-key inferior-ess-mode-map (kbd "M--") 'ess-smart-S-assign)

;; Hot key C-S-m for pipe operator in ESS
(defun then_R_operator ()
  "R - %>% operator or 'then' pipe operator"
  (interactive)
  (just-one-space 1)
  (insert "%>%")
```

```
  (just-one-space 1))
(define-key ess-mode-map (kbd "C-S-m") 'then_R_operator)
(define-key inferior-ess-mode-map (kbd "C-S-m") 'then_R_operator)

;; Truncate long lines
(add-hook 'special-mode-hook (lambda () (setq truncate-lines t)))
(add-hook 'inferior-ess-mode-hook (lambda () (setq truncate-lines t)))


(defun ess-rmarkdown ()
"Compile R markdown (.Rmd). Should work for any output type."
(interactive)
;; Check if attached R-session
(condition-case nil
    (ess-get-process)
  (error
   (ess-switch-process)))
(let* ((rmd-buf (current-buffer)))
  (save-excursion
    (let* ((sprocess (ess-get-process ess-current-process-name))
   (sbuffer (process-buffer sprocess))
   (buf-coding (symbol-name buffer-file-coding-system))
   (R-cmd
    (format "library(rmarkdown); rmarkdown::render(\"%s\")"
    buffer-file-name)))
      (message "Running rmarkdown on %s" buffer-file-name)
      (ess-execute R-cmd 'buffer nil nil)
      (switch-to-buffer rmd-buf)
      (ess-show-buffer (buffer-name sbuffer) nil)))))

(define-key ess-mode-map "\M-ns" 'ess-rmarkdown)

(defun ess-rshiny ()
  "Compile R markdown (.Rmd). Should work for any output type."
  (interactive)
  ;; Check if attached R-session
  (condition-case nil
      (ess-get-process)
    (error
     (ess-switch-process)))
```

```
  (let* ((rmd-buf (current-buffer)))
    (save-excursion
      (let* ((sprocess (ess-get-process ess-current-process-name))
     (sbuffer (process-buffer sprocess))
     (buf-coding (symbol-name buffer-file-coding-system))
     (R-cmd
      (format "library(rmarkdown);rmarkdown::run(\"%s\")"
      buffer-file-name)))
(message "Running shiny on %s" buffer-file-name)
(ess-execute R-cmd 'buffer nil nil)
(switch-to-buffer rmd-buf)
(ess-show-buffer (buffer-name sbuffer) nil)))))

(define-key ess-mode-map "\M-nr" 'ess-rshiny)
```

6 Elpy for python programming

```
(elpy-enable)
(setq elpy-rpc-python-command "python3")
(setq python-shell-interpreter "python3")

;; ipython
(elpy-use-ipython "python3")

;; Enable company
(add-hook 'python-mode-hook 'company-mode)
(add-hook 'inferior-python-mode-hook 'company-mode)
```

7 AUCTeX for LaTeX programming

```
(load "auctex.el" nil t t)

;; Appearance
(require 'font-latex)
(set-face-attribute 'font-latex-math-face nil :foreground "#ffffff")

(setq TeX-auto-save t)
(setq TeX-parse-self t)
```

```
;; Enable query for master file
(setq-default TeX-master nil)
(setq TeX-save-query nil)
(setq TeX-PDF-mode t)
(setq font-latex-fontify-sectioning 'color)
(setq font-latex-fontify-script nil)

;; Word-wrap
(add-hook 'TeX-mode-hook (lambda () (setq-default word-wrap t)))

;; Completion
(require 'company-auctex)
(company-auctex-init)
```

8 Web Developments

```
;; Rainbow mode
(add-hook 'html-mode-hook 'rainbow-mode)
(add-hook 'css-mode-hook 'rainbow-mode)
```

9 Polymode

```
(require 'polymode)
(require 'poly-R)
(require 'poly-markdown)
(add-to-list 'auto-mode-alist '("\\.md" . poly-markdown-mode))
(add-to-list 'auto-mode-alist '("\\.Snw$" . poly-noweb+r-mode))
(add-to-list 'auto-mode-alist '("\\.Rnw$" . poly-noweb+r-mode))
;; (add-to-list 'auto-mode-alist '("\\.Rmd$" . poly-markdown+r-mode))
(add-to-list 'auto-mode-alist '("\\.Rmd$" . poly-markdown-mode))
(add-to-list 'auto-mode-alist '("\\.rapport$" . poly-rapport-mode))
(add-to-list 'auto-mode-alist '("\\.Rhtml$" . poly-html+r-mode))
(add-to-list 'auto-mode-alist '("\\.Rbrew$" . poly-brew+r-mode))
(add-to-list 'auto-mode-alist '("\\.Rcpp$" . poly-r+c++-mode))
(add-to-list 'auto-mode-alist '("\\.cppR$" . poly-c++r-mode))
```

10 Pdf-tools

```
(pdf-tools-install)
(setq pdf-view-display-size "fit-page")
(setq auto-revert-interval 0)
(setq ess-pdf-viewer-pref "emacsclient")
(setq TeX-view-program-selection '((output-pdf "PDF Tools")))
```

11 Magit

```
;; Currently magit cause some error when auto revert mode is on
(setq magit-auto-revert-mode nil)
```

12 Draft

Settings in this section are not yet organized but are being used