A short introduction to GNU Emacs for the developer

Thomas Claveirole

EPX presentation, October 11, 2004
http://www.epita.fr/˜epx/

Copying this document

Introduction

Why learn all those aggressive shortcuts and weird tips?

- To go faster.
- Concentrate on the main task: coding.
- Working in a friendly environment usually means better results.
- Do not feel limited by your tools.

Before starting

```
cd
rm .emacs
cp ../.env/.emacs .
```

When required to change your configuration files, edit `.myemacs`.

Conventions

- `C-x` stands for Control x.
- `M-x` stands for Meta x or Escape x (those are equivalent).
- `x y` stands for x followed by y.
- Thus, `C-h i` stands for Control h, followed by i.
- `M-x` commands: read a command from the minibuffer.
- `M-x gnus` means: type `M-x` and `gnus` in the minibuffer, then press Enter.
- The extra combo: `M-C-X` for Meta-Control-Shift-x.

Moving around in a hurry

- The classics:
  - C-n, C-p: next line, previous line.
  - C-f, C-b: forward char, backward char.
  - C-v, M-v: scroll forward, scroll backward.
- Useful and often ignored:
  - M->, M-<: beginning/ending of file.
  - C-a, C-e: beginning/ending of line.
  - C-←, C-→: move word by word.

Cutting and pasting: the mark and the region

- `C-SPC`, `C-@`: Set the mark.
- `C-x C-x`: Interchange mark and point.
- `C-w`: Kill region.
- `M-w`: Save region as last killed text without actually killing it.
- `C-y`: Yank last killed text.
- `M-y`: Replace text just yanked with an earlier batch of killed text.

Alternative: `M-x transient-mark-mode`.

Multiple windows editing

- Multiple windows:
  - `C-x 2`, `C-x 3`: split windows.
  - `C-x 0`, `C-x 1`: close windows.
  - `C-x o`: other window.
  - `C-M-v`, `C-M-V` (mind the uppercases): scroll other window.

- Multiple frames:
  - `C-x 5 2`: new frame.
  - `C-x 5 0`: close frame.
  - `C-x 5 o`: change frame.
  - `C-x 5 f`: open file in new frame.

Multiple buffers editing

- `C-x C-f` opens files but does not close already opened buffers.
- `C-x b`: switch buffer.
- `C-x 5 b`: switch buffer other frame.
- `C-x k`: kill buffer.
- Setting up the `iswitchb` mode:
  - `M-x customize-group`, then enter group `iswitchb`.
  - Toggle Iswitchb mode to "on".
  - Mind Iswitchb regex, set and save you configuration.

Customizations

```
(global-font-lock-mode t)           ;; Syntax highlighting.
(scroll-bar-mode nil)               ;; Hide scroll bar.
(tool-bar-mode nil)                 ;; Hide tool bar.
(menu-bar-mode nil)                 ;; Hide menu bar.
(mouse-wheel-mode)                  ;; Use wheel.
(display-time)                      ;; Display time.
(setq display-time-24h-format)      ;; Clock format.
```

Miscellaneous editing tips

- Uniquify file names
  - `M-x customize-group`, group `uniquify`.
  - Just play with variables.
  - Especially look at Uniquify Buffer Name Style. I recommend "`post-forward-angle-brackets`".

- Automatically add newlines at EOF
  - `M-x customize-group`, group `editing-basics`.
  - Set the require-final-newline variable.

Other editing tips

- Delete trailing whitespaces
  - Use `M-x delete-trailing-whitespace`.
  - To do it automatically: add those lines to your initialization file:

```
(defun my-hook-save ()
  (cond ((not (eq major-mode 'message-mode))
         (delete-trailing-whitespace))))
(add-hook 'write-file-hooks 'my-hook-save)
```

- Automatic updates of copyrights
  - Add “`(add-hook 'write-file-hooks 'copyright-update)`” in your *GNU Emacs* initialization file.
  - Run `M-x customize-group`, group `copyright` and set Copyright Query.

Still talking about editing tips...

- Etags
  - Use `etags *.[ch]` to build tags (or `emacs-etag`).
  - Then you may use `M-.` (find tag) and `M-x complete-tag`.
  - To bind `M-'` to `M-x complete-tag` in C mode:

```
(defun my-etags-shortcuts ()
  (cond ((eq major-mode 'c-mode)
         (global-set-key (kbd "M-,") 'complete-tag))))
(add-hook 'c-mode-hook 'my-etags-shortcuts)
```

- The speedbar
  - Type `M-x speedbar` and play!
  - Some people find it very useful, and some others completely useless, so just have a try and make your own opinion.

Compilation

- `M-x compile` the first time...
- `M-x recompile` ... then.
- `` C-x ` ``: goto next error.
- Advantages
  - No need to go into a terminal. So no mouse moves or extensive shortcuts to circulate windows.
  - Going faster on the errors.
  - Colored output much more easier to read.
- Drawbacks Could not find any, do you?

Keyboard macros

- `C-x (`, `C-x )`: start/stop defining a macro.
- `C-x e`: execute the most recent macro.
- `M-x apply-macro-to-region-lines`: apply macro to region lines.
- `M-x global-set-key`: bind a key to a command.
- `M-x local-set-key`: bind a key to a command, for the current buffer only.
- "`(global-set-key "C-o" 'goto-line)`" in your *Emacs* initialization file: bind key `C-o` to function `goto-line`.

Predefined useful macro while in C mode

- `C-c C-c`: comment region.
- `C-u C-c C-c`: uncomment region.
- `M-/`: complete word.
- `C-c C-\`: add a backslash at the end of region lines.
- `C-u C-c C-\`: remove backslashes at the end of region lines.
- `M-a`, `M-e`: Next/previous statement.
- `C-M-a`, `C-M-e`: Next/previous block.

HS minor mode

- `M-x hs-minor-mode`
- `C-c @ M-C-h`: Hide all blocks.
- `C-c @ M-C-s`: Show all blocks.
- `C-c @ C-s`: Show block.
- `C-c @ C-h`: Hide block.

To have it automatically in C mode:

```
(add-hook 'c-mode-hook 'hs-minor-mode)
```

Grep'ing and finding text

- The classics:
  - `C-s`, `C-r`: forward/backward search.
  - `M-C-S`, `M-C-R` (mind the uppercases): forward/backward regular expression search.
- The killer features:
  - `M-x occur`: find occurrences of a word and report it as `M-x compile`.
  - `M-x grep`, `M-x grep-find`: grep files and report occurrences in the same way as `M-x compile`.

No more needs for a terminal [1/2]

The `dired` mode:

- `C-x d` or `M-x dired` to launch it.
- `Enter` to edit files or go into directories.
- `>` and `<` to navigate between directories.
- `R`, `C`: rename/move, copy.
- `d`, `x`: mark file for deletion, delete marked files.
- `q` to leave directory.

No more needs for a terminal [2/2]

- Reading manual pages
  - `M-x man`: open a man page using man.
  - `C-h i` or `M-x info`: browse the info pages.
- Running gbd
  - `M-x gdb`

No more needs for a terminal

- When a term is absolutely needed:
  - `M-x ansi-term`: open an ANSI terminal.
  - Advantages
    - No window handling using the window manager.
    - All emacs shortcuts are effectives.
    - The terminal is just another buffer. Therefore it can be killed, switched, . . .

Conclusion

- Lots of GNU *Emacs* features have been learned.
- Using *Emacs* should be a lot more fun now.
- But there are still lots of things to discover...
  - You could print and use *The* GNU Emacs *reference card*.
  - For advanced knowledge you should look at *The* GNU Emacs *manual*.

References

[Lrd] *Emacs* tricks web page from the LRDE wiki.
www.lrde.epita.fr/cgi-bin/twiki/view/Projects/EmacsTricks.

[Gnu] The GNU *emacs* manual.
www.gnu.org/software/emacs/manual.

[Ema] The GNU *emacs* reference card.
http://etudiant.epita.fr/~clavei_t/mini-emacs/refcard.ps.