

<https://clx.asso.fr/spip/?J-ai-debute-emacs>



J'ai débuté emacs...

- Documentations - Laboratoire -



Date de mise en ligne : mardi 25 février 2003

Copyright © Club LinuX Nord-Pas de Calais - Tous droits réservés

Cet article s'adresse aux débutants et à ceux qui, comme moi il n'y a pas si longtemps, ont un peu peur d'emacs.

Mes débuts avec emacs

Document destiné aux débutants...

Comment j'ai (re)découvert emacs ?

Tout le monde connaît emacs... J'entends par là que ceux qui ont installé un GNU/Linux sur leur ordinateur sont tombés un jour ou l'autre sur cet éditeur. Je l'ai très vite laissé de côté car pas assez windows-like, et je rencontrais alors des personnes soit avec une complète aversion pour emacs soit passionnées et exaltées par celui-ci. Bref emacs ne laisse pas indifférent.

Après deux ans sans emacs, j'ai participé à un mardi Linux à Dunkerque présentant LaTeX. Cette soirée était animée par Georges Khaznadar et Denis Bitouzé (que je remercie particulièrement). Composant moi-même mes documents avec LaTeX, je fus assez surpris que Georges et Denis utilisent tous deux emacs... La démonstration faite ce soir-là me permit de voir la puissance d'emacs.

Dois-je rappeler que emacs est sous licence GPL ? (pour voir la licence taper Control-h puis Control-c dans l'éditeur).

Mon espace de travail â€” Installation

Je travaille avec une distribution Peanut allégée et le gestionnaire de fenêtres ctwm. Ma distribution ne contenant pas emacs, j'ai donc pris les paquets rpm de la distribution Mandrake8.0 et ai installé les paquets suivants :

- emacs-X11-20.7-17mdk ;
- emacs-20.7-17mdk ;
- auctex-11.13-1.

Je n'ai rien fait de spécial à l'issue de l'installation.

J'accède à emacs, en tapant `emacs` dans une fenêtre xterm.

Que vois-je ?

[<https://clx.asso.fr/spip/local/cache-vignettes/L367xH400/emacs-4-eae3a.png>]

On aperçoit :

- une barre d'outils (je ne l'utilise quasiment jamais) ;
- une zone de texte avec déjà des lignes dedans ;
- un ascenseur à gauche ;
- une barre de couleur en bas qui donne des informations sur le mode employé, etc. ;
- une zone de saisie en bas de cette barre de couleur : c'est le << mini-buffer >>.

Quelques notations : TRÈS IMPORTANT

emacs utilise les touches Control et Alt à profusion, je vais donc noter :

- C : la touche Control ou Ctrl ;
- M : la touche Alt (il est assez usuel de la noter ainsi... M=Meta) ;
- Esc : la touche Échap ;
- Ret : la touche Entrée ;
- Backspace : la touche d'effacement en arrière (située à côté du signe =) ;
- Tab : la touche de tabulation (située à gauche du A) ;

Avec ces notations :

- C-x : désigne un appui simultané sur les touches Control et x ;
- C-x C-f : désigne un appui simultané sur les touches Control et x suivi d'un appui simultané sur les touches Control et f, on pourra simplement laisser enfoncée la touche Control et presser successivement sur x et f.
- C-x b : désigne un appui simultané sur les touches Control et x, puis d'un appui sur la touche b (les touches Ctrl et x n'étant plus enfoncées).

Premières commandes

Comme on vient de l'entre-apercevoir, emacs n'aime pas l'imprécision dans la frappe au clavier ; retenir les raccourcis-claviers que nous allons voir permet de s'affanchir totalement de la souris et de gagner en rapidité. Aussi, je ne parlerai pas du tout (!) de la barre d'outils.

Le programme emacs n'est pas facile à prendre en main. Voici donc quelques raccourcis de base. Bien sûr cette liste n'est pas exhaustive. Il s'agit des combinaisons de touches que j'utilise régulièrement...

Manipulation des fichiers

- Ouvrir un fichier : C-x C-f ;
- Fermer un fichier : C-x k puis choix du fichier à fermer ;
- Créer un fichier : il suffit d'ouvrir avec C-x C-f puis de compléter avec le nom que vous voulez lui donner ;
- Enregistrer un fichier ouvert : C-x C-s ;
- Enregistrer avec un nom différent : C-x C-w ;
- Quitter emacs : C-x C-c .

Couper/Copier/Coller

- 2 solutions pour sélectionner : on sélectionne à la souris ou on marque l'emplacement de départ avec C-Barre d'espace puis on déplace le curseur avec les flèches de direction ;
- Tout sélectionner : C-x h ;
- Copier : M-w ;
- Couper : C-w ;

– Coller : C-y .

Chercher/Remplacer

- Recherche après le curseur : C-s ;
- Recherche avant le curseur : C-r ;
- Remplacement : M-% (attention cela revient en fait à presser simultanément les touches Alt-Shift-ù) puis répondre aux questions posées par emacs .

Se déplacer dans un texte

Les flèches de direction permettent de se déplacer simplement, mais on pourra utiliser ces touches avec Control...

Les touches Home et End (à côté de Inser et Suppr) permettent de positionner le curseur au début et à la fin du fichier

édité et les touches de pagination... paginent !

On utilisera surtout ces deux choses :

- Début de ligne : C-a ;
- Fin de ligne : C-e .

Effacer du texte

Les touches Suppr et Backspace remplissent les rôles attendus. On aura aussi recours à deux/trois choses simples :

- Effacer le mot avant le curseur : M-Backspace ;
- Effacer le mot après le curseur : C-d ;
- Effacer toute la ligne après le curseur : C-k ;

Buffers ou fichiers à€” Ouverture

Un buffer et une zone de texte en cours d'édition et qui n'est pas forcément sauvegardé sur le disque dur. Par exemple,

au démarrage, la zone de saisie est en partie remplie par trois lignes commençant par

```
This buffer is for notes you don't want to save ...
```

N'en tenez pas compte et ouvrez par exemple le fichier .bashrc :

Taper C-x C-f puis commencer par écrire .bas. Enfin, appuyer sur Tab, emacs complète alors avec un h puis s'arrête (il a trouvé plusieurs correspondances possibles pour ce début !), presser de nouveau la touche Tab. La fenêtre

se sépare en deux et emacs vous propose une liste contenant .bashrc, .bash_profile etc.

Soit vous complétez correctement en tapant le reste, soit vous appuyer sur la touche r puis Tab et emacs complétera le c

lui-même, soit vous cliquez avec le bouton du milieu sur le fichier à ouvrir.

Pour créer un fichier, il suffira de l'ouvrir à l'endroit voulu (C-x C-f puis de saisir nom du fichier dans le bon répertoire) ;

tant qu'il restera vide ou tant que vous ne l'enregistrerez pas explicitement, il sera considéré comme un buffer

c'est-à-dire
n'existant que pour emacs.

Ainsi on parle plus volontiers de buffers que de fichiers, les buffers étant l'ensemble des zones de texte ouvertes par emacs. Notons qu'à l'ouverture, il y en a pas mal (au moins *Messages* et *scratch*); cela déroute toujours au début mais après on s'y fait ;-)

Les trucs qu'il faut connaître

- Je me suis planté et je veux annuler : C-_ (Control et la touche 8) ;
- Je suis paumé dans ma commande Control bidule avec Alt Machin et tout semble bloqué : C-g ;
- Je suis paumé dans ma commande Control bidule avec Alt Machin et tout semble bloqué (bis) : Esc-Esc-Esc ;
- Il m'a coupé la fenêtre en deux et j'arrive plus à enlever celle du bas : C-x 1 (C-x puis Shift-&) ;
- J'ai ouvert deux fichiers en même temps... Il est où le premier ? : C-x C-b (donne la liste des buffers) ;
- Appuyer sur Tab permet deux choses suivant le cas : l'autocomplétion dans le minibuffer ou la ré-indentation correcte de la ligne en cours ;
- C-z iconifie emacs... rapide et efficace ;
- Je veux de l'aide : C-h t donne accès à un tutoriel. Le mien est en anglais :-)
- Je veux de l'aide (bis) : C-h m donne accès aux raccourcis accessibles dans le buffer en cours.

Les apports de Auctex : le mode LaTeX

Pour bénéficier du mode Auctex, il faut avoir cette ligne-là dans votre fichier .emacs (cf. section suivante) :

```
(require 'tex-site)
```

Cela fait déjà beaucoup mais voilà encore plus... C'est pour la bonne cause ;-)

Voilà quelques raccourcis **INDISPENSABLES** pour utiliser LaTeX efficacement et sans effort (enfin presque) :

- Ouvrir un environnement : C-c C-e ;
- Ouvrir un attribut de police : C-c C-f C-b pour du texte gras (b pour *bold*) ou C-c C-f C-i pour de l'italique, etc.
- Commenter une région sélectionnée : C-c ;
- Décommenter ... : C-c ;
- Fermer un environnement ouvert à la main (ça ne m'arrive jamais :-p) : C-c] (Control c puis AltGr-) ;
- Dans un environnement à item, taper M-Ret et non pas Ret tout seul : un nouvel \item s'ouvre !
- Compiler : C-c ;
- Visualiser après compilation : C-c (facile !!).

Pour en savoir plus, n'hésitez pas à aller voir dans la barre d'outils dans laquelle vous trouverez un menu LaTeX qui propose différents sous menus comprenant des commandes et presque à chaque fois les raccourcis-claviers correspondants.

Notons aussi une petite possibilité très pratique : le changement d'environnement avec C-u C-c C-e ... Absolument génial :-) merci Auctex.

J'avoue avoir beaucoup de choses à découvrir encore avec Auctex...

Personnaliser emacs : le fichier .emacs

J'ai débuté emacs...

Ce fichier situé normalement dans votre répertoire personnel donc `~/.emacs` sert à personnaliser votre environnement emacs en modifiant quelques paramètres fixés par défaut ou en ajoutant des fonctionnalités qui n'existent pas. Ce fichier est évalué au démarrage d'emacs.

Pourquoi personnaliser... Parce que certains trucs ne me satisfaisaient pas. En voici une liste avec leur solution à placer dans le fichier `.emacs` (si ce n'est pas déjà fait) :

– Toute alerte est sanctionnée par un bruit atroce qui réveille ma fille pendant son dodo !

```
(setq visible-bell 't)
```

À la place de ce petit tintement pénible, vous verrez clignoter les deux lignes aux extrémités verticales de la fenêtre... C'est la première chose à faire pour personnaliser emacs.

– L'indentation est pas top, on doit souvent la rectifier en appuyant sur tab :-(

```
(fset 'indenterTout "\C-xh\234\C-g") (global-set-key "\C-ci" 'indenterTout)
```

Maintenant en appuyant sur C-c i tout s'indente tout seul !

– Je veux indenter comme les pros et emacs le fait pas :-(

```
;; ce qui suit est un code integralement rédigé par Olivier Ramare ;; apres ma question sur le forum idoine
: fr.comp.applications.emacs (defun only-spaces-on-line-before nil "t if point is between
beginning-of-line and first non-whitespace character, nil else. nil if point is at beginning of line."
(let (res (beg (line-beginning-position))) (save-excursion (save-restriction
(narrow-to-region beg (line-end-position)) (skip-syntax-forward " ") ; beware:
linefeed/newline are whitespaces (setq res (if (= (point) beg)
nil (= (current-indentation) (- (point) beg)))) (widen)) res)) (defun
my-backward-indentation nil (interactive) (if (not (only-spaces-on-line-before)) (delete-char
-1) (let ((foundp nil) (cc (current-indentation)) (ncc) (save-excursion (while (and (not
(bobp)) (not foundp)) (forward-line -1) (beginning-of-line) ; for the eobp to work
(setq foundp (> cc (setq ncc (current-indentation)))))) (save-restriction
(narrow-to-region (line-beginning-position) (line-end-position)) (skip-syntax-forward " ")
(if (not foundp) (delete-char (- cc)) (delete-char (- ncc cc)) (widen))))
(global-set-key [return] 'newline-and-indent) (global-set-key [backspace] 'my-backward-indentation)
```

Bon... J'y comprends pas grand-chose mais je vous explique ce que cela fait :

Ce code redéfinit les touches Ret et Backspace pour qu'elles prennent en compte les niveaux d'indentation antérieurs. Cela permet de toujours bien indenter en appuyant sur Ret et de toujours bien se positionner sur une indentation en appuyant sur Backspace... Bref le comportement idéal.

L'inconvénient, c'est qu'en redéfinissant la touche Ret, on ne peut plus valider quoi que ce soit dans le minibuffer ! On ne peut donc même plus ouvrir un fichier ! Il faut donc dans le minibuffer non plus valider avec la touche Ret mais avec C-m (cela demande un peu de gymnastique au début mais on s'y fait et j'en suis satisfait).

– Comment naviguer simplement entre les fichiers ouverts ?

J'ai débuté emacs...

```
(require 'iswitchb) (iswitchb-default-keybindings)
```

On peut naviguer simplement avec C-c b (et aussi toujours avec C-c C-b mais qui split la fenêtre :-)

– Super emacs ;-) mais alors les couleurs :-)

On trouve sur emacsfr.org une solution simple à ce problème... avec des possibilités de thèmes !

Pour ma part, j'ai simplement inséré ceci dans mon .emacs :

```
(set-foreground-color "grey80") (set-background-color "black")
```

– Sauver l'historique des commandes passées au minibuffer parce que c'est ch*ant de devoir tout retaper !!!

```
(require 'save-history)
```

Ben oui c'est tout simple !!! Maintenant on peut récupérer ce qui a été tapé précédemment avec les flèches haut et bas. Essayer avec l'ouverture de plusieurs fichiers (cela remplace par exemple le menu << récents >> qu'on voit souvent).

Bizarre... Chez moi j'ai adapté le contenu de .emacs et je n'ai pas de changement de comportement

Il doit y avoir une erreur de syntaxe dans le fichier et cela n'a pas empêché emacs de démarrer... aller voir le buffer << journal >> du chargement d'emacs : *Messages*.

Un shell dans mon emacs !

Essayez M-x shell... Pour cela, tapez M-x puis le minibuffer attend... Et il suffit de saisir `shell` puis de valider (appuyer sur Ret ou C-m) et là extase :-))

Et puis, on se prend à rêver, on tape `mc` :

```
Your terminal lacks the ability to clear the screen or position the cursor.
```

Et là on retombe mollement :-((alors essayez :

M-x ansi-term (n'oubliez pas que la touche Tab permet de compléter les commandes longues) et là... `mc` fonctionne :-)) mais bon je vous le déconseille pour deux raisons :

- les raccourcis-claviers de `mc` ne marchent pas (ben ouais !)
- emacs sait gérer les fonctionnalités basiques de manipulations des fichiers : M-x dired .

Je n'ai pas été plus loin dans M-x dired... A vous !

Je veux faire mes macros emacs mais le lisp et moi ça fait trois !

Vous aussi ? Bon ben écoutez bien la manip :

- Commencer une macro (tout ce que vous ferez au clavier sera mémorisé par emacs) : C-x (;
- Terminer une macro : C-x) ;
- Donner un nom à la macro : M-x name-last-kbd-macro ;
- La << rejouer >> : M-x insert-keyboard-macro

(Pour les deux dernières commandes, n'oubliez pas d'utiliser l'auto-complétion permise par un appui sur la touche Tab).

La dernière commande affiche sur le buffer courant le code Lisp de votre macro, il suffit alors de faire un couper/coller dans votre fichier .emacs !

C'est comme cela que j'ai fait ma fonction C-c i pour indenter automatiquement.

Ce qui ne va pas ... chez moi

- chez moi emacs est en anglais ;
- modifier par menu interactif la config de AucTex insère dans le .emacs plein de choses pas très compréhensible ;
- la coloration syntaxique n'est pas au top, j'ai dû rater un truc encore que...

Dernière remarque : apprendre emacs permet de mieux utiliser la console (avec bash par exemple). En effet certains raccourcis sont communs... A tester !

Conclusion

Voilà donc mes débuts avec emacs, je voulais vous faire partager cette découverte car j'ai tout simplement abandonné les autres éditeurs tellement emacs m'a séduit. Les raccourcis m'ont parus très pénibles au départ mais je les considère maintenant comme un avantage sur les autres éditeurs. emacs déborde de fonctionnalités à découvrir, il permet de personnaliser à l'extrême son environnement d'édition.

emacs peut aussi servir de logiciel de messagerie (avec le module gnus) ou encore de navigateur Web. Emacs est disponible aussi dans une version plus orientée vers X : Xemacs ; bref emacs sait presque tout faire.

Netographie

En anglais

- <http://www.gnu.org/software/emacs/>
- <http://www.schacherer.de/frank/tech...>
- <http://www.refcards.com/>
- <http://www.nongnu.org/auctex/>

En français

- <http://www.emacsfr.org/>
- <http://www.linux-france.org/article...>
- <http://parinux.org/ressources/mini-...>

Post-scriptum :

Bien évidemment cet article réunit beaucoup de choses apprises << sur le tas >> et n'a pas la prétention d'être exhaustif, beaucoup de commandes n'ont pas été citées. Je vous invite à m'envoyer vos combinaisons de touches indispensables ou encore à compléter cet article.