



Edit MACroS - EMACS

Votre éditeur de la mort pour
Prologin

Dernière Révision : 30/04/2002

Sommaire

INTRODUCTION	1
QUELQUES NOTIONS UTILES	2
LES BINDINGS.....	2
LES BINDINGS DE BUFFER.....	2
LES BINDINGS DE FRAMES.....	3
LES BINDINGS DE RECHERCHE / NAVIGATION.....	3
LES BINDINGS DE MACRO	4
SAUVER LA MACRO SUR DISQUE :	5
AUTRES BINDINGS SYMPA.....	5
COMPILER SOUS EMACS.....	6

Introduction

EMACS est un éditeur multi-usages, que nous utilisons pour faire des tonnes de choses, depuis notre mail jusqu'à nos programmes, en passant par les news et les documents.

Il est configuré au lancement par deux fichiers : le .emacs et le .myemacs, qui, si nous en avons le temps, seront places dans chacun de vos comptes. Il s'agit du légendaire .emacs du Titi précité et d'un .myemacs perso, qui devrait vous éviter de trop vous paumer dans les touches prévues à la base, en offrant un vrai delete, un vrai PageUp, PageDown, BackSpace, Home, End, etc...

Mais, si EMACS dispose de tonnes de raccourcis clavier, ce n'est pas pour amuser la galerie. Y'a une raison.

Ils font plein de choses top.

Mais avant d'entamer les raccourcis clavier ("*bindings*")...

Quelques Notions Utiles

Buffer	Zone mémoire ou un texte est stocke.
Frame	Fenêtre Emacs (il peut en contenir des tas, imbriqués dans tous les sens, au premier plan, derrière...) permettant de visualiser un buffer.
Binding	Raccourci clavier. On les note de deux façons : C-... : Ctrl+... M-... : Meta+... C'est-à-dire Esc/Alt+... La seule différence c'est que Esc est bloquante : on peut la relâcher avant de presser la touche en combinaison avec. Par contre on doit absolument taper ensuite la commande. Exemple : C-g (Ctrl+G) annule la requête en cours
Minibuffer	Barre noire où apparaissent des infos quand on fait une commande quelconque (en bas de EMACS)
Barre de statut	La barre blanche en bas d'un frame. Elle contient plusieurs zones, aussi appelées champs, que voici :
	Zone 1 "----" Buffer sauvé "--*-" Buffer modifié non sauvé "--%%-" Buffer en lecture seule
	Zone 2 "Emacs". Supeeeeeer.
	Zone 3 Nom du fichier
	Zone 4 "(Mode)" (Mode = C, Pascal, HTML, Prolog, Lisp...)
	Sur les chouettes confs on rajoute :
	Zone 5 ligne/colonne
	Zone 6 "Bot" -> Le bas (Bottom) du buffer est visible "Top" -> Le haut du buffer est visible "xx%" -> On est à xx % du buffer (Ctrl+L = actu.)

Les bindings

Les Bindings de Buffer

NOTA : Si vous voulez la totale, Menu "Help", "List keybindings"...

Comment gérer vos buffers ? Facile. On dispose, entre autres, des bindings suivants :

(Nota : C-x est bloquant aussi)

Binding	Action
C-h	Aide.
C-k	Coupe la fin de la ligne dans le presse-papiers
C-o	Insère une ligne vide
C-u n cmd_name	Répète n fois la commande cmd_name
C-x C-f	Ouvre/Crée un fichier
C-x C-s	Sauve le buffer
C-x C-b	Liste les buffers. Clic du milieu sur le buffer voulu.
C-x i	Insère un fichier dans le buffer

Les Bindings de Frames

Pour gérer l'affichage, le refresh et la navigation entre vos frames, voici un petit B-A-BA :

Binding	Action
C-l	Rafraîchit l'écran (couleurs, barre de statut...)
C-x o	Buffer suivant (liste circulaire de buffers ouverts)
C-x 0	Frame courant passe en arrière-plan
C-x 1	Frame courant occupe tout l'avant-plan
C-x 2	Divise le frame courant dans le sens horizontal
C-x 3	Divise le frame courant dans le sens vertical
C-x k	Détruit le buffer (laisse le frame)

Les Bindings de Recherche / Navigation

Dans un buffer, encore faut-il pouvoir se déplacer, rechercher du texte, copier, couper, coller...

Binding	Action
C-Space	Pose une marque de début de région (la position du curseur est toujours la fin de la région)
C-a	Début de ligne
C-e	Fin de ligne
M-a	Début de bloc
M-e	Fin de bloc
M-b	Un mot en arrière
M-f	Un mot en avant
C-d	Efface le caractère sur le curseur
BackSpace	Idem mais avant le curseur
C-v	Page suivante
M-v	Page précédente
C-s	Recherche d'une chaîne en sens avant
C-r	Recherche idem en arrière
M-<	Début du buffer
M->	Fin du buffer

Binding	Action
C-w	Coupe la région courante dans le p.-p.
M-w	Copie la région courante dans le p.-p.
C-y	Colle le presse-papiers

NOTA : Si vous disposez du .myemacs voulu, vous avez les touches suivantes, mappées comme sur PC :

Touche	Action	Au lieu de
Home	Début de ligne	Début de buffer
End	Fin de ligne	Fin de buffer
PgUp	Page précédente	
PgDn	Page suivante	
Delete	Efface sur le curseur	Efface avant (comme BackSpace)
F9	Compilation	Minimise fenêtre
F2	Sauvegarde buffer	

Les Bindings de Macro

Pour DEFINIR une macro, on ouvre la définition avec :C-x (On fait son taf, puis on termine avec :C-x) Et on a "Keyboard macro defined".

* DEFINIR une région sur laquelle appliquer la macro : on a vu C-Space au début de la région puis déplacement du curseur jusqu'à la fin de la région. La, on fait :

```
M-x apply-macro-to-region-lines
```

i.e. M-x appl[tab] -> complétion automatique, comme dans le shell. Et hop ! Goal.

Exemple :

Indenter toute une région de 4 caractères sur la droite...

1) Définir la macro. On suppose être sur une ligne à modifier. On va faire début de ligne, 4 x espace, puis flèche bas. On y va :

```
C-x ( C-a <espace> <espace> <espace> <espace> <Down> C-x )
```

2) On se place sur le haut de la zone à indenter (on aura donc pris soin d'enregistrer la macro sur la première ligne à indenter, qui n'a plus besoin de faire partie de la région maintenant ;). On marque le début avec :

```
C-Espace
```

3) On se place sur la fin de la région avec le curseur. On applique :

```
M-x appl <tab> <return>
```

4) Gagné !...

Pigé ? La suite...

NOMMER une macro pour l'utiliser à la suite d'un M-x :

```
M-x name-last-kbd-macro (utilisez la complétion, hein...) <return>  
Nom_de_la_macro <return>
```

Attention : Ce nom n'est plus valable à la prochaine session.

SAUVER la macro sur disque

```
C-x C-f Nom_Buffer_Receveur_Macro <return>  
M-x insert-kbd-macro <return>  
Nom_macro (la complétion marche) <return>  
C-x C-s (pour sauver le buffer)
```

Comment croyez-vous qu'on crée un .myemacs ?!

CHARGER la macro depuis le disque

```
M-x load-library <return>  
Nom_fichier_lib <return>
```

Autres Bindings Sympa

On notera :

Binding	Action
M-x justify-line	Justifie la ligne en cours de traitement
M-x center-line...	
M-x center-region-lines	Beaucoup mieux.
M-x capitalize-region	Met toutes les initiales des mots en Maj. !

et plein d'autres encore...

Révision : faire une macro pour justifier toute une région ?

Réponse :

```
C-x ( <Home> M-x justify-line <return> <Down> C-x )  
(Placement en haut de la région)
```

C-Espace

(Placement en bas de la région)

M-x apply-macro-to-region-lines <return>

Compiler sous EMACS

Pour compiler, la solution simple consiste à activer la fenêtre de shell, taper ou reprendre la commande (gcc/gpc/etc...), valider, attendre, et suivant les cas repasser sur Emacs pour corriger ou exécuter.

Bon.

M-x compile, anyone ?

Cette commande (bindée sur F9 dans les bonnes confs, sinon : M-x global-set-key <return> <F9> compile <return>) permet de lancer une ligne de commande complète de compilation. Vous ne pouvez toutefois pas utiliser d'alias ou autres astuces dans cette ligne, ce doit être un appel à des exécutables fournis par le système, ou des commandes Shell.

Cette commande vous ouvre de surcroît une fenêtre de compil, et les résultats de la compil sont donc accessibles dans un buffer (baptisé par ailleurs "**Compilation**"). Voilà, voilà... Assez sympa.

Pour exécuter, de toute façon, c'est dans le shell, c'est clair...

A propos : M-x shell c'est rigolo aussi mais attention ne pas abuser ;-)