

# Une courte introduction à Emacs

Thomas TEMPÉ

Le 30 janvier 2001

## Introduction

Emacs est l'éditeur de texte de toutes les légendes. Il est l'oeuvre de Richard Stallman (initiateur de la Free Software Foundation (FSF) et du projet GNU) et de nombreux contributeurs. Il est d'abord apparu en 1975 au MIT, comme un ensemble de macros pour l'éditeur Teco (aujourd'hui disparu), d'où son nom, contraction de «*Editing Macros*».

Aujourd'hui, Emacs est un projet titanesque, dont les installations représentent couramment de l'ordre de 50Mo de code. Il est construit autour d'un interpréteur de E-Lisp (une variante du langage fonctionnel Lisp) écrit en C. L'éditeur lui-même est très largement écrit en E-Lisp, et sa conception modulaire a permis le développement d'un grand nombre de modules (modes, applications et autres) écrits en E-Lisp. Emacs emploie également largement les expressions régulières.

Il existe plusieurs branches de développement de Emacs. La plus connue et celle que nous utiliserons s'appelle GNU Emacs, qui en est actuellement à la version 21.3. La seconde plus connue, également disponible dans la plupart des distributions de Linux, s'appelle X-Emacs (qui n'a rien à voir avec le support X-Window. Les deux peuvent fonctionner avec ou sans support X. Lancer Emacs avec l'option `-nw` pour qu'il s'exécute sans support X). Il existe aussi une ribambelle de versions développées pour certaines plates-formes, comme Windows ou AmigaOS...

## 1 Ce qu'Emacs est, et n'est pas

Emacs n'édite que des fichiers texte, sans formatage particulier. Impossible de souligner ou de mettre en gras. Cette tâche revient à des traitements de texte (OpenOffice.org) ou des processeurs typographiques (LaTeX).

Les fichiers texte (typiquement appelés «fichiers ASCII», mais qui peuvent être écrits avec divers jeux de caractères) sont utilisés dans un très grand nombre d'application, et Emacs peut s'interfacer avec des outils comme les compilateurs, shells, processeurs de documents, ou correcteurs orthographiques.

Emacs est en fait plus qu'un éditeur de texte. C'est un environnement de travail complet. Il gère de multiples documents (buffers), de multiples fenêtre, offre une interface utilisateur cohérente, et peut contenir des applications élaborées, comme des clients mail, ftp, news, ou web.

## 2 Première prise en main

Démarrons Emacs avec la commande `emacs truc`, ce qui a pour effet de créer le fichier `truc`, et de l'ouvrir. L'écran est divisé en quatre zones : les menus déroulants (accessibles avec F10), la zone de saisie, la barre d'état, et le mini-buffer (dernière ligne).

La barre d'état indique si le fichier en cours a été modifié (\*\*) ou est accessible en lecture seule (%%), son nom, le mode d'édition, et la ligne en cours.

Un nouveau fichier, `truc`, a été créé par Emacs. Tapons quelques lignes de texte. Emacs n'est pas un éditeur modal, comme Vi, ce qui le rend moins effrayant au premier abord.

Sauvons maintenant nos modifications. En maintenant la touche `contrôle` enfoncée, appuyons successivement sur `X` et sur `S`. Un message s'affiche dans le mini-buffer. Le fichier est sauvé.

## 3 Les raccourcis clavier

Les premières versions d'Emacs ont été développées avant même l'apparition des interfaces graphiques. L'interface utilisateur principale est le clavier. Son apprentissage permet d'interagir avec le programme beaucoup plus efficacement qu'avec une souris.

Les raccourcis clavier sont obtenus en gardant enfoncée soit la touche `contrôle` (C-x), soit la touche `alt`, appelée *meta* (M-x) (La touche *meta* est parfois autre chose que `alt`, notamment dans les versions Windows de Emacs, ou elle est remplacée par `echap`). Voyons ici une première liste de raccourcis utiles :

- C-x C-s sauve le fichier en cours ;
- C-x C-w sauve sous un autre nom ;
- C-x C-f *find file* : ouvre un nouveau fichier, ou le crée s'il n'existe pas ;
- C-x i insère le contenu d'un fichier existant à la position du curseur ;
- M-x saisit une commande dans le mini-buffer.

Le mini-buffer est utilisé pour inter-agir avec Emacs, par exemple pour taper le nom d'un fichier à ouvrir ou à sauver. On peut aussi y taper le nom de commandes Emacs. Les raccourcis clavier Emacs ne font en fait qu'appeler les commandes qui y sont liées. Par exemple, on peut aussi sauver le fichier en cours avec M-x `save-buffer`.

## 4 Comment se déplacer dans le texte

Emacs est optimisé pour l'édition de fichiers très volumineux. Comme les développeurs ont mieux à faire de leur temps que de garder le doigt sur les touches de direction en attendant que le texte défile, il existe des moyens de se déplacer rapidement dans le texte :

- **Page Up**, **Page Down** déplacent le curseur d'une page vers le haut ou le bas ;
- **Home**, **End** l'amènent au début ou à la fin du fichier ;
- **les flèches de direction**, C-f, C-b, C-p, C-n déplacent le curseur d'une lettre ou d'une ligne ;
- **C-flèches**, M-flèches, M-f, M-b, M-p, M-n le déplacent d'un mot ou d'un paragraphe ;

- **C-a**, **C-e**, **M-a**, **M-e** le mettent au début et à la fin de la ligne, ou du paragraphe en cours;
- **C-l** rafraîchit l'écran, et place la ligne portant le curseur au milieu.
- **C-v**, **M-v** affichent la vue (page) suivante et précédente. Si deux fenêtres sont visibles à la fois, C-M-v fait défiler le texte dans la fenêtre inactive.

Ne négligez pas, à terme, d'apprendre les raccourcis clavier basés sur f (forward), b (back), p (previous), et n (next). Ils peuvent vous faire gagner un temps considérable.

Notez également que souvent, les mêmes combinaisons de touches utilisées avec C- et M- font des choses similaires.

## 5 La recherche

La recherche de texte est également un bon moyen de se déplacer dans un long texte, car elle est très rapidement accessible par les raccourcis clavier.

**C-s** et **C-r** permettent de faire une recherche incrémentale (en avant et en arrière), c'est-à-dire au fur et à mesure que l'on saisit. On peut aussi faire des recherches par expressions régulières (**M-x search-forward-regexp**, **M-x search-backward-regexp**). La touche Entrée termine la recherche, et l'appui répété sur C-s ou C-r la reconduit autant de fois que nécessaire.

**M-x replace-string** permet de remplacer une chaîne avec une autre, et **M-x query-replace-string** demande une confirmation avant chaque remplacement.

## 6 Facilités d'édition

Le mini-buffer bénéficie d'un système de complétion automatique, et d'une historique. Par exemple, tapez **M-x y Tab Tab** pour voir la liste des commandes commençant par y. Si vous saisissez "o" puis Tab à nouveau, la commande "yow" s'affiche en entier. Entrée l'exécutera. **M-x flèche vers le haut** rappelle la commande précédente. Ceci est aussi valable pour les noms de fichiers, ou pour les phrases à remplacer.

Pour arrêter l'action en cours, on utilisera **C-g**. On peut aussi répéter une commande autant de fois qu'on le veut avec **C-u nombre** ou **M-nombre**. Par exemple, **C-u 9999 C-b** a de fortes chances de vous ramener au début du fichier. **C-u 80 \*** vous remplira une ligne de petites étoiles.

Il est possible d'annuler une commande ou modification avec **C-/** (**M-x undo**). On peut ainsi revenir en arrière de manière presque illimitée, sur les modifications qui ont eu lieu dans la session courante.

## 7 Le copier-coller

**C-d**, **M-d**, **M-backspace** suppriment la lettre ou le mot suivant ou précédent, et **C-k** et **M-k** suppriment la ligne ou le paragraphe suivant.

Toute suppression de plusieurs lettres à la fois sauve automatiquement ces lettres dans ce que l'on appelle le *kill ring*. Il est alors possible de les rappeler avec **C-y** (*yank*), et de consulter tout le kill ring avec **M-y**. Le kill ring contient par défaut les vingt derniers morceaux de texte supprimés.

Il existe aussi des “registres”, ou presse-papier nommés, auxquels on peut accéder avec **copy-region-to-register** ou **C-x r x**, et **insert-register** ou **C-x r g**. Voir le paragraphe sur la marque.

## 8 Les modes

Saisissez une longue phrase sans retour à la ligne. Vous constaterez qu’elle est affichée sur plusieurs lignes, mais pas coupée en deux. Ceci est intéressant pour la programmation, où les retours à la ligne peuvent avoir un sens, mais moins pour les textes, où les longues lignes peuvent être tronquées par certains programmes à l’affichage. Tapez maintenant **M-x auto-fill-mode**. Le mot “fill” s’ajoute dans la barre d’état. Saisissez à nouveau une longue ligne. Remarquez comme elle est coupée en deux quand elle devient trop longue.

Revenez au début de la première ligne, et ajoutez quelques lettres. La ligne n’est pas re-formatée. Pour forcer le re-formatage du paragraphe en cours, tapez **M-q**.

Il édite deux types de modes, les modes majeurs et les modes mineurs. On n’est jamais que dans un mode majeur à la fois, qui dictera le comportement général de l’éditeur. On peut par contre activer plusieurs modes mineurs simultanément. Chacun modifiera l’un ou l’autre aspect du mode majeur.

Il est possible de changer de mode en tapant dans le mini-buffer, le nom du mode suivi de “-mode”. Par exemple, **M-x c-mode** fait passer en mode C. Le fait d’ouvrir un fichier .c fait le même effet. Le mode C surligne les parenthèses que l’on referme, et indente proprement la ligne en cours lorsqu’on appuie sur Tab. On peut désactiver un mode mineur en re-tapant son nom dans le mini-buffer.

## 9 Les buffers et les fenêtres

Chaque fichier ou texte de documentation est représenté par un buffer. Emacs peut ouvrir plusieurs buffers simultanément (**M-x list-buffers**). On peut passer d’un buffer à un autre avec **C-x b**, puis en tapant ou choisissant le nom du buffer voulu. **C-x s** permet de sauver tous les buffers ouverts. **C-x k** ou **kill-buffer** ferme le buffer en cours.

Chaque buffer peut être affiché dans une ou plusieurs fenêtres. Pour créer une deuxième fenêtre, tapons **C-x 2**. Une deuxième barre d’état apparaît au milieu de l’écran. On a maintenant 2 zones de saisies, permettant de visualiser le même fichier. Chaque ensemble { zone de saisie, barre d’état } représente une fenêtre. Les modifications apportées dans l’une se reportent dans l’autre. **C-x o** permet de passer de l’une à l’autre. On peut maintenant changer le buffer visible dans chacune des fenêtres, pour éditer deux fichiers dans le même écran. On peut aussi scinder chaque fenêtre en deux, horizontalement avec **C-x 2** ou verticalement avec **C-x 3**. On re-collera 2 fenêtres ensemble avec **C-x 0**. Pour revenir à un affichage avec une seule fenêtre, on tapera **C-x 1**. Enfin, on peut modifier la taille des fenêtres avec **M-x enlarge-window** et ses amis.

## 10 La marque

La marque est une sorte de signet qui peut être placé n'importe où dans un buffer, avec le raccourci **C-espace**. On peut échanger la position de la marque et du curseur avec **C-x C-x**. La marque est aussi placée automatiquement à la position actuelle lorsqu'on appuie sur `home`, `end`, lorsqu'on fait une recherche ou qu'on colle (`yank`) un morceau de texte. Ceci permet par exemple de revenir rapidement là où on en était quand on se déplace au début du fichier par erreur (fréquent sur les claviers de portables).

La zone comprise entre le curseur et la marque est appelée une région. De nombreuses commandes Emacs s'appliquent à la région, notamment **C-w** et **M-w**, qui coupent et copient respectivement la région dans le kill ring.

## 11 Les macros clavier

Les macros sont des suites de touches que l'on enregistre, puis que l'on peut répéter plusieurs fois. On démarre l'enregistrement avec **C-x** ( ou **M-x start-kbd-macro** ). On réalise ensuite une action, par exemple, rechercher (`C-s`) la prochaine occurrence du mot "cyber-papy", et la remplacer par "concombre masqué". On arrête ensuite l'enregistrement avec **C-x** ) ou **M-x end-kbd-macro**. On peut maintenant l'exécuter à nouveau, soit avec **C-x e** soit avec **call-last-kbd-macro**. On peut aussi taper **name-last-kbd-macro**, et saisir un nom pour notre macro (par exemple "concombre"). On pourra alors l'appeler avec **M-x concombre**. Attention cependant, les macros ainsi créées ne sont pas sauvegardées automatiquement.

## 12 La correction orthographique

Elle n'est pas proposée directement par Emacs, mais par `ispell` (ou `aspell`), qui sont des outils génériques de correction orthographiques typiquement installés par défaut sur toute machine Linux. Voir la documentation de `ispell` pour installer les bons dictionnaires par défaut.

Les commandes **ispell-word**, **ispell-region**, **ispell-buffer** permettent de corriger un document en cours. On peut aussi utiliser le mode mineur **M-x flyspell-mode** pour réaliser la correction au vol (comme sous Word;). `ispell` a cependant parfois du mal avec les accents...

## 13 À l'aide !

Emacs possède un système de documentation assez impressionnant. Vous pourrez consulter la documentation de plusieurs manières, en général à partir du raccourci **C-h** suivi d'une lettre. Les fenêtres d'aide contiennent des liens hypertexte, qui permettent un minimum de navigation.

- **C-h ?** donne la liste des rubriques d'aide ;
- **C-h a** permet d'effectuer une recherche par mots-clef dans les noms de commandes ;
- **C-h c** nomme la fonction appelée par un raccourci particulier, et
- **C-h f** donne la description d'une fonction dont on connaît le nom ;

- **C-h w** donne directement la description complète de la fonction appelée par un raccourci...
- **C-h l** montre les 100 dernières touches appuyées;
- **C-h m** affiche la documentation des divers modes majeurs et mineurs actuels.

Les personnes intéressées trouveront également de l'aide sur les différentes méthodes de saisie (pour les jeux de caractères étrangers) un tutoriel, et une liste des différentes façons de dire “bonjour” dans le monde.

## 14 Les commandes utiles ou amusantes

Nous avons maintenant fait le tour des concepts les plus importants de Emacs. On pourrait en citer d'autres, comme les abréviations, les packages, ou l'agenda (diary). Il ne vous faudra que peu de temps pour vous familiariser avec elles.

La coloration syntaxique est proposée par le **M-x font-lock-mode**. Il est aussi possible de recourir à la complétion automatique lors de la saisie de texte avec **M-x dabbrev-expand** et **M-x dabbrev-completion**, disponibles avec les raccourcis **M-/** et **C-M-/**.

Emacs sauvegarde automatiquement les fichiers ouverts dans des fichiers tels que **truc~** ou **#truc#**. Il est possible de récupérer un tel fichier avec **M-x recover-file**. Emacs se transforme en gestionnaire de fichiers quand on lui demande d'ouvrir un répertoire. Et pour les malvoyants, on peut utiliser *emacspeak* pour prononcer le texte affiché.

Emacs permet également d'envoyer du courrier électronique (pour peu qu'un serveur mail soit installé sur la machine) avec **C-x m**. L'application Gnus (**M-x gnus**) est un client usenet complet. Il existe aussi un client mail intégré, un client ftp (**ange-ftp**), un support CVS, un mode permettant d'exécuter un shell dans une fenêtre emacs (**M-x shell**), des commandes permettant la compilation (**M-x compile**), un calendrier (**M-x calendar**) permettant de connaître la date dans le calendrier étiopien, chinois ou inca **M-x calendar-chinese-date-string**. Enfin, le jeu de la vie est disponible avec **M-x life**, les citations de Zippy avec **M-x yow**, le psychologue avec **M-x doctor**, et la rencontre des grands esprits avec **M-x psychanalyze-zippy**.

## Conclusion

Emacs est un outil fantastique. Nous n'avons ici qu'effleuré la surface de ses fonctionnalités. Peu de gens peuvent prétendre connaître entièrement Emacs; mais avec quelques jours d'entraînement, on arrive très vite à enrichir son répertoire de raccourcis, et à gagner en productivité. Au fur et à mesure, on découvre de nouvelles fonctionnalités, de nouvelles commandes, de nouveaux raccourcis, et on intègre de nouveaux réflexes qui, à chaque étape, nous font gagner quelques secondes ou fractions de seconde par opération. Finalement, couplé avec des outils comme LaTeX ou gcc, Emacs permet de rédiger un compte-rendu ou de déboguer un programme beaucoup plus rapidement, et surtout avec un bien plus grand confort d'utilisation.

Emacs est donc l'un de ces outils qui deviennent peu à peu indispensables, irremplaçables, et font paraître fade toute alternative.