

[Download as PDF](#)

# Emacs et Vi

---

Objectifs:

- Savoir utiliser un éditeur de texte
- Savoir utiliser les éditeurs de texte Emacs et Vi

Éléments techniques abordés:

- *Les commandes*
  - emacs, vi

## Introduction

---

Un éditeur de textes est un programme qui vous permet de créer et de modifier des fichiers de données. Unix propose principalement deux éditeurs de texte : **vi** et **emacs**.

## L'éditeur emacs

---

En réalité, **emacs** est plus qu'un simple éditeur de texte; il s'agit d'un environnement de travail complet formant une surcouche d'Unix. Depuis emacs, il est possible non seulement de créer et modifier des fichiers texte, mais également de jouer, envoyer et recevoir du courrier, accéder à des forums de discussion, manipuler des fichiers et des répertoires et aussi concevoir des programmes.

## La touche <Ctrl>

emacs fait appel à un nombre important de combinaisons de touches, bien plus qu'il n'est possible de mémoriser. La première chose à faire est de comprendre comment **emacs** gère le clavier. Comme tous les éditeurs de texte, **emacs** gère les touches normales (lettres de l'alphabet, chiffres, caractères de ponctuation, etc) mais aussi deux touches supplémentaires : <Ctrl> et <Méta>. La touche <Ctrl> s'utilise de manière usuelle, c'est-à-dire en la maintenant appuyée pendant que vous pressez une autre touche. Ainsi, la commande **<Ctrl>-h** qui permet de disposer de l'aide interactive s'obtient en pressant <Ctrl> et h. La notation **emacs** pour la touche <Ctrl> est la lettre C. La nouveauté avec emacs, c'est que certaines commandes réclament plusieurs séquences <Ctrl> d'affilée, ou une séquence <Ctrl> suivie d'une lettre. En voici deux exemples: Pour quitter emacs, il faut entrer **C-x C-c**, c'est à dire presser **C-x** puis **C-c**. Pour lancer le didacticiel intégré, il faut entrer **C-h t**, soit **C-h** puis la lettre **t**.

## La touche <Meta>

En plus de la touche <Ctrl>, emacs utilise une autre touche spéciale, appelée la touche META. Cette touche s'emploie de la même manière que les touches <Ctrl> ou <Shift>, c'est-à-dire en la maintenant enfoncée pendant que vous entrez un autre caractère. La notation emacs pour indiquer une combinaison de touches faisant intervenir la touche <Meta> est similaire à celle utilisée pour <Ctrl>, excepté le fait que le caractère **M** remplace le caractère **C**. Pour indiquer la combinaison <Meta> **a**, il faudra donc lire **M-a**. Par exemple, pendant l'édition d'un texte, la commande qui permet de descendre d'une page est **C-v** et celle qui permet de remonter d'une page est **M-v**. En regardant sur votre clavier, vous vous êtes sûrement aperçus qu'il

n'existe pas de touche <Meta>, cette touche ayant été enlevée des claviers que nous utilisons actuellement. Cette touche peut être remplacée par la touche <Alt> ou encore par la touche <Echap> ou <Esc>. La différence avec la touche <Esc> par rapport à la touche <Meta> est qu'il faut pour cette méthode, presser <Esc> puis la relâcher avant de presser la seconde touche.

## Noms de touches spéciales

Dans emacs, quelques notations particulières identifient des touches spéciales, comme le montre le tableau suivant. En plus des touches <Ctrl> et <Meta>, vous pourrez rencontrer des combinaisons faisant intervenir simultanément <Ctrl> et <Meta>.

| Nom        | Touche                |
|------------|-----------------------|
| <b>C-</b>  | <Ctrl>                |
| <b>M-</b>  | <Meta>                |
| <b>DEL</b> | <Delete>              |
| <b>ESC</b> | <Escape>              |
| <b>LFD</b> | <Linefeed> (<Ctrl-J>) |
| <b>RET</b> | <Return>              |
| <b>SPC</b> | <Escape>              |
| <b>TAB</b> | <Tab>                 |

## Comment lancer l'éditeur emacs

Pour lancer emacs, il suffit d'entrer la commande **emacs**. La syntaxe de base est:

```
emacs [fichier]
```

où **fichier** est le nom du fichier que vous souhaitez éditer.

Pour activer emacs sans fichier, il suffit de saisir simplement la commande:

```
emacs
```

## Quitter emacs

Pour sortir d'**emacs**, utilisez la commande **C-x C-c**. S'il n'y a pas de fichier à sauvegarder, emacs vous avertit avec le message: (No files need saving) (Aucun fichier à sauvegarder) et vous retournez au prompt de l'interpréteur de commandes.

Si vous n'avez pas sauvegardé tous vos fichiers, **emacs** vous propose de le faire avant de sortir. Pour chaque fichier non sauvegardé, **emacs** affiche son nom et propose une liste de choix.

Voici un exemple. Supposons que vous avez travaillé sur un fichier nommé document qui n'a pas été sauvegardé. Si vous pressez **C-x C-c**, vous obtiendrez le message: Save file /usr/harley/document ? (y, n, !, ., q, C-r or C-h) (Sauvez le fichier /usr/harley/document ? (y, n, !, ., q, C-r or C-h))

**emacs** vous demande si le fichier doit être sauvegardé. Vous avez alors plusieurs choix. **y** correspond à oui (yes) et indique que vous voulez sauvegarder, **n** correspond à non (no) et indique que vous ne voulez pas sauvegarder vos dernières modifications.

Pour sauvegarder tous les fichiers ouverts d'un coup, pressez **!**. **q** permet de quitter emacs sans sauvegarder davantage de données. Pour enregistrer le fichier courant et sortir tout de suite après, pressez **..** Si vous essayez de sortir alors que tous les fichiers ne sont pas tous sauvegardés, **emacs** affichera le message:

```
-----
Modified buffers exist ; exit anyway? (yes or no)
(Buffer modified ; Voulez-vous sortir tout de meme?)
-----
```

Vous devez alors rentrer soit **yes** en entier, soit **no**. Un buffer peut être considéré comme un espace de travail propre.

Lorsque vous travaillez avec un fichier unique, **emacs** affiche celui-ci directement sur l'écran. Si vous éditez plusieurs fichiers simultanément, **emacs** vous demandera quoi faire pour chaque fichier ouvert et il se peut très bien que vous ayez oublié le contenu d'un des fichiers. Si cela vous arrive, **emacs** vous demande alors que faire de ce fichier. Pressez **C-r** pour demander à **emacs** de vous montrer le fichier, de manière à pouvoir prendre la décision appropriée. Pour cesser de consulter le fichier, il faut entrer **q**.

## Les buffers

**emacs** a pour principale qualité de vous permettre de faire plusieurs choses en même temps. Vous pouvez ainsi éditer autant de fichiers que vous le souhaitez, et passer de l'un à l'autre selon vos besoins.

Pour que ce soit possible, **emacs** gère un espace de travail distinct, un BUFFER (un espace mémoire propre) pour chaque tâche. Si vous éditez trois fichiers distincts, **emacs** les stockera dans trois *buffers* mémoire séparés, un par fichier.

Tout ce que vous voyez et écrivez dans **emacs** est conservé dans un *buffer*. Par exemple, **emacs** contient une aide intégrée que vous pouvez consulter quand bon vous semble. Lorsque vous pressez **-h**, **emacs** crée un buffer pour contenir ces informations d'aide.

Pour garder trace de tous les buffers, **emacs** affecte un nom unique à chacun d'eux. Lorsque vous débutez l'édition d'un fichier, **emacs** crée un buffer avec le même nom que celui du fichier. En éditant un fichier nommé **document**, vous créez en fait un buffer **document** contenant ce fichier.

### En résumé :

- Tout ce que vous faites dans **emacs** est stocké dans un buffer.
- Chaque buffer a un nom unique.
- Vous pouvez créer un buffer quand bon vous semble.
- Vous pouvez détruire un buffer quand bon vous semble.
- Certains buffers sont créés par vous, d'autres par **emacs** lui-même.

## Fenêtres

Donc, tout ce que vous faites est stocké dans un buffer. Comment alors peut-on visualiser le contenu d'un buffer? Pour ce faire, **emacs** crée une ou plusieurs fenêtres et affiche dans chaque fenêtre le contenu d'un buffer. Un écran emacs peut ainsi se décomposer en plusieurs fenêtres, chacune affichant le contenu d'un buffer. **emacs** vous permet aussi d'afficher le contenu d'un buffer dans plusieurs fenêtres simultanément, ce qui peut être utile lorsque le buffer est trop volumineux pour tenir dans une seule fenêtre. Vous pouvez aussi ouvrir plusieurs fenêtres pour examiner différentes parties d'un même fichier en même temps.

## Aller plus loin avec emacs

Un memento sur les combinaisons de touches les plus courantes est donné sur ce wiki. Un tutoriel est également disponible à l'adresse <http://www.tuteurs.ens.fr/unix/editeurs/emacs.html> [<http://www.tuteurs.ens.fr/unix/editeurs/emacs.html>].

### Exercice 1 (Emacs)

1. Copiez le fichier `/usr/share/doc/emacs23-common/BUGS` dans votre home directory. Le chemin complet est long à taper, heureusement la touche Tab permet de compléter automatiquement les noms des fichiers et des répertoires quand il n'y a pas ambiguïté.
2. Changez vos droits sur ce fichier en `rw-`.
3. Lancez Emacs en tapant la commande: `emacs &`. Si vous avez oublié le `&`, tapez `Ctrl-z` dans la fenêtre Terminal pour mettre Emacs en pause, suivi de la commande `bg` pour le mettre en tâche de fond.
4. Ouvrez le fichier `NEWS` que vous venez de copier (`Ctrl-x Ctrl-f`).
5. Déplacez votre curseur jusqu'au début de la ligne 501 (`Esc-x goto-line`).
6. Sélectionnez (`Ctrl-Space`) le texte jusqu'à la ligne 583.
7. Coupez le texte sélectionné (`Ctrl-w`).
8. Rendez-vous ensuite à la fin du document (`Esc - >` en mode texte, `Ctrl-End` en graphique) et collez le texte précédemment coupé (`Ctrl-y`).
9. Retournez au début du document (`Esc - <` en mode texte, `Ctrl-Home` en graphique) et faites une recherche (`Ctrl-s`) du mot Emacs. Vous pouvez passer d'un mot au suivant en faisant à nouveau `Ctrl-s`.
10. Allez une fois de plus au début du document et remplacez (`Esc-%`) toutes les instances de Emacs par XEmacs. Emacs vous demande de confirmer le remplacement de chaque mot en appuyant sur y. Vous pouvez forcer le remplacement de tous les mots sans confirmation en appuyant sur la touche !.
11. `Ctrl-k` permet de couper la fin d'une ligne à partir de l'emplacement du curseur, sans avoir à utiliser `Ctrl-Space + Ctrl-w`. Répété plusieurs fois, `Ctrl-k` permet de couper également les lignes sous le curseur. Comme avant, `Ctrl-y` permet de coller le texte qui vient d'être coupé. Essayez.
12. Sauvez votre nouveau fichier.
13. Quittez Emacs.

## L'éditeur de texte vi

Vi (prononcer "Vi-äie") est un éditeur de texte visuel (ne pas confondre avec graphique). Il a été écrit par William Joy en 1976 pour une des premières versions de la distribution BSD d'Unix.

Vim (pour Vi IMproved et qui se traduit par : Vi amélioré) est directement inspiré par Vi, mais contrairement à ce dernier, Vim est un logiciel libre. La première version a été écrite et diffusée par Bram Moolenaar en 1991.

Sur les machines de travaux pratiques de l'UFR, c'est Vim qui est installé, aussi dans la suite, on utilisera indifféremment l'appellation Vi ou Vim.

La syntaxe pour lancer Vi est la suivante :

```
vi nom_du_fichier
```

ou

```
vim nom_du_fichier
```

Une fois le fichier ouvert, il vous est possible de vous déplacer à l'aide des curseurs, ainsi que des touches h, j, k et l (au cas où le clavier ne posséderait pas de curseurs fléchés). Si le fichier n'existe pas, il sera créé.

## Les modes de Vi

Vi possède 3 modes de fonctionnement :

- **Le mode normal:** celui dans lequel vous êtes à l'ouverture du fichier. Il permet de taper des commandes
- **Le mode insertion:** Ce mode permet d'insérer les caractères que vous saisissez à l'intérieur du document. Pour passer en mode insertion, il suffit d'appuyer sur la touche Insert de votre clavier, ou à défaut sur les touches *i*
- **Le mode de remplacement:** Ce mode permet de remplacer le texte existant par le texte que vous saisissez. Il vous suffit de réappuyer sur insert (ou *ï*) pour passer du mode insertion au mode remplacement, et d'appuyer sur la touche Echap pour revenir en mode normal

## Les commandes de base

| Commande         | Description  |
|------------------|--|
| :q               | Quitte l'éditeur (sans sauvegarder)  |
| :q!              | Force l'éditeur à quitter sans sauvegarder (même si des modifications ont été apportées au document) |
| :wq              | Sauvegarde le document et quitte l'éditeur   |
| :file <i>nom</i> | Sauvegarde le document sous le nom spécifié  |
| i                | Passage en mode insertion  |
| <Echap>          | Quitter le mode insertion  |

## Les commandes d'édition

| Commande | Description  |
|----------|--|
| x        | Efface le caractère actuellement sous le curseur                               |
| dd       | Efface la ligne actuellement sous le curseur                                   |
| dxd      | Efface x lignes à partir de celle actuellement sous le curseur                 |
| nx       | Efface n caractères à partir de celle actuellement sous le curseur             |
| x»       | Indente x lignes vers la droite à partir de celle actuellement sous le curseur |
| x«       | Indente x lignes vers la gauche à partir de celle actuellement sous le curseur |

## Aide

| Commande              | Description                    |
|-----------------------|--------------------------------|
| :help                 | Aide de Vi                     |
| :help <i>commande</i> | Aide de Vi sur <i>commande</i> |

## La recherche et le remplacement

Pour rechercher un mot dans un document, il vous suffit (en mode normal) de taper / suivi de la chaîne à

rechercher, puis de valider par la touche entrée. Il est alors possible d'aller d'occurrence en occurrence grâce à la touche n.

Pour remplacer une chaîne de caractère par une autre sur une ligne, il existe une commande très puissante sous Vi utilisant les expressions régulières. Pour plus d'informations concernant les expressions régulières, vous pouvez vous référer au site <http://www.commentcamarche.net/contents/php/phpreg.php3> [http://www.commentcamarche.net/contents/php/phpreg.php3]. Voici sa syntaxe :

```
:s/chaîne_a_replacer/chaîne_replacante/g
```

Il est possible de la généraliser à tout le document grâce à la syntaxe suivante :

```
:%s/chaîne_a_replacer/chaîne_replacante/g
```

## Aller plus loin avec Vi

Un mémento Vi se situe sur ce site : <http://www.ens.math-info.univ-paris5.fr/stage-unix/doku.php?id=mementos:vi> [http://www.ens.math-info.univ-paris5.fr/stage-unix/doku.php?id=mementos:vi]

## Exercice 2 (Vi)

1. Créez un fichier que vous nommerez testvi.txt
2. Passez en mode Insertion et saisissez le texte suivant : « C'est facile de travailler avec vi ». Revenez en mode commande.
3. Modifiez « C'est facile » en « C'est très facile ». Revenez en mode commande.
4. Ajoutez à la fin du texte sur une nouvelle ligne : « mais je préfère écrire sous Word. ». Revenez en mode commande
5. Supprimez le mot « Word », passez en mode Insertion et le remplacez par « ... oups ! OpenOffice ».
6. Sauvegardez votre fichier et quittez vi .

tp/tp5.txt · Dernière modification: 2015/08/14 17:31 par janiszek