

vim : éditeur de texte

- Objet : vim
- Niveau requis : [débutant, avisé](#)
- Commentaires : *Administrer la rédaction des fichiers de configurations, et plus encore, les pages web et autres...*
- Débutant, à savoir : [Utiliser GNU/Linux en ligne de commande, tout commence là !](#) 😊
- Suivi : [à-compléter](#)
 - Création par [smolski](#) le 10-07-2009 19:37:07
 - Testé par [smolski](#) le 10-07-2009 19:37:07
- Commentaires sur le forum : [Lien vers le forum concernant ce tuto](#)¹⁾

Introduction

Vim est un éditeur de texte particulièrement puissant et rapide.

Son auteur voulait au départ faire un clone libre de [vi](#), l'éditeur de base sous UNIX, mais tant de nouvelles fonctionnalités ont été ajoutées que l'on peut dire maintenant qu'il s'agit d'un programme véritablement original.

Installation

Ben, comme d'habitude :

```
apt-get update && apt-get install vim
```

Cela installe 2 paquets.

Dire oui à la demande et hop ! C'est fait...

Pour un vim complet :

```
apt-get update && apt-get install vim-gtk exuberant-ctags
```

vim-nox

Voilà un vim débarrassé de tout ce qui se rapporte au serveurX !

```
apt-get update && apt-get install vim-nox vim-scripts exuberant-ctags
```

Et vous pouvez le mettre en éditeur par défaut pour tous avec la commande :

```
update-alternatives --set editor /usr/bin/vim.nox
```

Trôôp cool... Merci de ces indications captfnfab !

Vimtutor

Les débutants rêvent d'un tuto avec des exemples à tester, vim l'a fait !

Comme le dit **Melodie** :

- *Le meilleur des tutoriels est déjà là près de vous... Et vous savez quoi... il s'est installé d'office sur vot' pc avec votre install de **vim**²⁾ via le paquet vim-runtime !³⁾*

- *Noooooon ?*

- *Et si mon gars, d'office dans l'office... Tout prêt à vous servir !*

Pour le voir, dans la console saisissez et validez :

```
vimtutor
```

et... laissez-vous guider !

Vous vous retrouverez sur une page en **vim** que vous pouvez modifier en suivant les exercices de base proposés.

De plus... c'est en français, ouaiiiiiiiiiiiiiiiiiis ! 😊

À consommer sans modération....

Ou faites des essais en créant des répertoires avec [mkdir](#) et des fichiers avec [touch](#), puis lancez-vous pour écrire, vous déplacer, copier, supprimer... tout en ligne de commande **vim**.

Les modes

Vim se gère sous 3 modes : le mode normal, le mode saisie et le mode commande.

Le mode commande simple

Après l'ouverture d'un fichier :

```
vim /chemin/de/votre/fichier
```

Permet le déplacement par les flèches du clavier, la suppression... Mais pour écrire et remplacer, il faut passer en mode saisi par la touche `ij` ou sur la touche `Insert`

À l'ouverture, nous pouvons nous déplacer dans le texte du fichier avec les flèches du clavier où avec les lettres du clavier G h j k l jusqu'à l'endroit où nous désirons intervenir.

- La touche h pour vous déplacer vers la gauche.
- La touche l pour vous déplacer vers la droite.
- La touche j pour vous déplacer vers le bas.
- La touche k pour vous déplacer vers le haut.

- La touche G pour vous déplacer de page en page
 - G pour aller à la dernière ligne
 - gg pour aller à la première ligne
 - 2g pour aller à la ligne n°2 ; 3g pour aller à la ligne n°3, etc.,

On peut aussi se déplacer avec les flèches ← ↓ ↑ →

Le mode commande "simple" c'est-à-dire le mode qui permet de lancer des commandes sans avoir à appuyer sur la touche `Return` permet d'utiliser des commandes pour se déplacer, mais aussi pour passer en mode saisie ou en mode commande ex

Le mode saisie

Le mode saisie s'obtient par la touche `i` (ou la touche `Insert`).

Le mode commande ex

Le mode commande permet d'enregistrer ce que l'on a écrit/modifié, mais aussi de quitter vim, ou de copier/coller, ou d'effacer, tout ça avec plusieurs options disponibles et plus encore... A découvrir.

Voir ici pour l'essentiel des commandes identiques à l'éditeur VI : [LES COMMANDES VI](#)
Ou un PDF à télécharger ici : http://wiki.chezlefab.net/_media/tuto_nix/vim/vimqrc-fr.pdf
De chez le **captnfab**. Hop !

Attention, en mode commande, les flèches du clavier ne permettent plus les déplacements habituels dans le texte. Il faut utiliser les codes de placement du curseur pour la saisie.
Utilisez les lettres du clavier G h j k l !

Pour continuer l'action précédente, taper sur :

- 1 - `Échap` pour revenir en mode commande,
- 2 - puis `:w` `Entrée` pour enregistrer
- 3 - et `:q` `Entrée` pour quitter vim.

Et voilà les modifications faites.

Les commandes précédées de deux points sont en fait des commandes de l'éditeur ex. vi et ex sont deux interfaces distinctes du même logiciel d'édition de texte. Et à sa base vim c'est vi !



Alors que vi est une interface orientée écran, ex est une interface orientée ligne. Toutes les commandes ex sont disponibles à partir de l'éditeur vi. Lorsqu'on appuie sur la touche ":", on passe en fait à l'interface ex (orientée ligne). Cela vous permet d'exécuter de nombreuses commandes de manipulation de fichiers sans quitter vi.

- Voir :

1. <http://archive.oreilly.com/pub/a/linux/excerpts/9780596529833/the-vi-ex-and-vi>

[m-editors.html](#)2. <http://www.computerhope.com/unix/uex.htm>

Ouvrir 2 fichiers

```
vim fichier1 fichier2
```

On ira d'un fichier à un autre en tapant les touches du clavier `:` + n `↵` Entrée et `:` + `⇧` Shift + n `↵` Entrée

Ouvrir 2 fichiers dans 1 terminal

Cela permet de copier en utilisant les commandes vim de l'un à l'autre.

```
vim fichier1
```

Puis en mode commande⁴⁾ :

```
:split fichier2
```

Et on ira d'un fichier à un autre par le mode commande en tapant 2 fois la touche w tout en gardant la touche `Ctrl` appuyée :

```
Ctrl + w + w
```

split 1 fichier

Vous éditez d'abord le fichier, puis vous tapez la commande :

```
:split
```

Ce même fichier s'ouvre alors sur 2 fenêtres !

Par exemple, on peut lire les lignes de variables dans une fenêtre et leur application dans l'autre.

Pour changer de fenêtre la même procédure s'applique : `Ctrl` + w + w

Fermeture

- La commande `:q` quitte entièrement vim
- La commande `:close` quitte la fenêtre active en conservant vim ouvert.
- La commande `:only` laisse ouverte la fenêtre active et ferme toutes les autres avec le message habituel si vous n'avez pas enregistré les modifications.

Fenêtres-dimensions

Ouvrir par exemple une fenêtre de 10 lignes se fait par la commande :

```
:10split fichier1
```

Pour modifier cette taille confortablement vous pouvez activer la souris :

```
:set mouse=a
```

et manipuler maintenant les barres de séparation avec le pointeur.

split vertical

split ouvre d'office des fenêtres partagées horizontalement.

Pour avoir des fenêtres partagées verticalement, la commande est *vsplit*, soit :

```
:vsplit fichier2
```

Bingo ! 😊

vimdiff

la commande **vimdiff** permet de comparer les différences entre les fichiers ouverts par pré-sélectionnement automatique.

La commande :

```
vimdiff fichier1 fichier2
```

diffget

L'option de la commande **diffget** permet de recopier entre les fichiers les différences pré-sélectionnées par vimdiff.

Procédure en exemple :

Éditez 2 fichiers tests écrits différemment l'un de l'autre :

```
vim diff1
```

et rédiger :

```
Bonjour,  
Comment allez-vous ?
```

Enregistrez et fermez et pour le second :

```
vim diff2
```

et rédiger :

```
Salut !  
Comment vas-tu ?
```

Les ouvrir simultanément avec vimdiff :

```
vimdiff diff1 diff2
```

Nous obtenons ce terminal qui ne manque pas d'allure :



En nous plaçant sur la ligne **Bonjour**, de diff1 nous ouvrons la commande

```
:diffget
```

et nous indiquons de prendre copie des différences contenues dans le fichier diff2 pour les porter dans diff1 ainsi :

```
:diffget diff2
```

En validant, le remplacement est immédiat et vimdiff n'indique plus aucune différence. On enregistre et on quitte chacun des fichiers par la commande :

```
:wq
```

Si au contraire on veut modifier à l'inverse, il faut se placer dans la fenêtre diff2 par la combinaison clavier tapée deux fois :

```
Ctrl + w Ctrl + w
```

Ce qui nous place dans la fenêtre suivante et refaire la commande mais à l'inverse :

```
:diffget diff1
```

Et voilà !

Nota

Nous avons ici en exemple deux fichiers du même répertoire, en cas de chemin de répertoires différents, il faut les indiquer totalement.

diffput

Non, ce n'est pas une commande d'insulte grivoise adressée entre fichiers discordants mais c'est celle qui permet à l'inverse de [diffget](#) de remplacer ou de copier les données du fichier actif vers l'autre fichier lorsqu'elles n'y sont pas présentes.

Remerciement à **cthuluh** qui via l'irc a guidé de ses longs bras mes petits doigts vers cette commande.

Tableaux vim

Tableaux divers d'utilisation variées :

- [vim en tableaux](#) Une toile de maître qui nous sera beaucoup enviée ! 😊

Vim - Les macro

Enregistrer pour réutilisation facile d'une édition de texte complexe faite en série, c'est ici :

- [macros](#)

Vim - La navigation

Pour connaître le chemin du fichier en cours d'édition, utiliser le raccourci :

`Ctrl` + g

Naviguer dans les répertoires sans quitter la commande vim

- [Parcourir les répertoires avec Vim](#)

Liens

- <http://www.commentcamarche.net/faq/7961-petit-tuto-sur-vi-vim>

Traduction de toute la documentation de vim dans langue de Molière (un boulot admirable ! 😊) :

- http://cfennajoui.net/vim/index.php?p_menu=presentation

Un formidable outil détaillant les principales commandes vim, venu tout droit du **captnfab** ! 😊

Voir son site :

- http://wiki.chezlefab.net/tuto_nix/vim/accueil

Voir aussi cet excellent site :

- <http://www.gcolpart.com/howto/vim.php4>

Le wiki français non officiel de vim :

- <http://vim-fr.org/index.php/Accueil>

Extension - Configurer vim

Tout un tas d'astuce de configuration des extensions vimiesques sur le forum :

- <https://debian-facile.org/viewtopic.php?id=27246>

TRAVAUX PRATIQUES

Pour ces exercices utilisez des répertoires et des fichiers que vous aurez créés dans le home user. Voir les commandes en mode terminal ici : [Commandes Linux](#)

```
test-1/essai-1.txt
```

et

```
test-2/essai-2.txt
```

Yop... Au boulot maint'nant !

TP-01 Sélection clavier - La lettre "v"

Vous pouvez sélectionner banalement les textes ou paragraphe avec la souris, mais... mais :

Depuis votre terminal ouvrez vim sur votre fichier `essai-1.txt`, soit :

```
vim test-1/essai-1.txt
```

Et s'il n'y a pas de texte..?

Bigre, taper sur la lettre `i` du clavier (pour insérer) et écrivez un texte court, beau, plaisant à voir et à modifier, un texte à la Prevert, une poésie libre de deux trois lignes en quelques mots...

Ou plus prosaïquement, la copie d'un fichier de configuration, genre `fstab` ou `.bashrc` ou encore le fichier default de `site-availability`... genre...

En mode commande (tapez `Echap` pour y parvenir), déplacez le curseur au début de la première ligne à intervenir (copier, déplacer, supprimer... voir les raccourcis commandes vim), tapez la lettre `v` du clavier et déplacez le curseur jusqu'au dernier mot où vous désirez intervenir...

Yop, la sélection s'opère sous nos yeux même ! Reste à appliquer ce que nous désirons faire.

Sans la souris, tout tôle clavier ! Trôôôôp facile...

C'est plus précis car visuel, qu'en mettant des numéros et/ou des nombres de lignes...

Merci à **TrueDD** sur l'irc #debian-facile, sans qui cette manip nous serait inconnu jusqu'à s'theure !

TP-02 Modification d'un terme en série - La commande ":s/"

*Toujours de ce cher bon **TrueDD** sur irc #DF... que la soie brossée de mille tapis d'Orient lui caresse la plante des pieds à jamais...*

Voici issue de mon fichier /etc/apache2/sites-available/default ce paragraphe pour exemple :

```
Alias /sitevideo/ "/media/disk-2/sitevideo/"
<Directory "/media/disk-2/sitevideo/">
  Options Indexes MultiViews FollowSymLinks
  AllowOverride None
  Order deny,allow
  Allow from all
  Allow from 127.0.0.0/255.0.0.0 ::1/128
</Directory>
```

Voilà, c'est un alias afin que sur mon serveur le *sitevideo* soit lu par url.

Je désire ajouter un nouveau site, un site jeu pour mes petits enfants (c'est du vécu, hein !) qui concerne un **resto**.

Il me faut d'abord copier ce paragraphe puis le coller dans le même fichier (c'est au plus simple), et changer la route **sitevideo** en **site-resto** cool.

En mode commande (taper sur Echap pour vous en assurer...) taper la lettre : v

et appliquez le **TP-01** pour sélectionner tout le paragraphe taper : y

Et pour le coller après, déplacer le curseur à l'endroit où vous voulez insérer le nouvel alias et taper ↑ Shift + p

Toujours en mode commande, positionner le curseur sur le A de *Alias* du début de ce paragraphe copié
(utiliser les lettres g h j k l commandes identiques aux flèches du clavier pour ce faire... hi hi hi).

Taper la lettre v et sélectionner tout le paragraphe au clavier (voir **TP-01**, encore...)

Lors donc, taper la commande

```
:s/sitevideo/site-resto/g
```

de vos petits doigts graciles, valider et hop ! Le remplacement, c'est tout fait.

Moi j'dis que plus facile que ça, jamais... Non ?

TP-03 Insertion d'un dièse (#) en série sur plusieurs lignes

Pour insérer un dièse (#) sur 4 lignes d'un coup afin de les commenter, après avoir déterminé le numéro de ligne de la première à commenter, voici la commande :

Par exemple, commenter de la 7ème à la 11ème ligne :

```
:7,11s/^/#/
```

ou bien :

```
:7,11s/^/#
```

Merci **Thom1** ! 😊

TP-03 Insertion de ligne

Pour insérer une ligne en dessous la ligne courante, depuis le mode normal de vim (tapez `Echap` d'abord pour en être sûr) tapez o (De même pour insérer une ligne au dessus de la ligne courante, en mode normal tapez O)

Vous voilà en mode Insertion dans une ligne vierge ajoutée.
Tapez la touche `Echap` pour revenir en mode Commande.

Astuces

Live

A dire vrai, j'ai toujours préféré le direct live genre sur **fstab** ou **interfaces**, fichiers que je manipule souvent et dont je connais bien les arcanes...
Et c'est que du bonheur ensuite...

Vimrc

Nous pouvons améliorer la fenêtre de vim par de petites touches personnelles dans le fichier `/etc/vim/vimrc` (si vous voulez appliquez les modifications à tous vos utilisateurs, sinon modifier le fichier `~/.vimrc`)

Par exemple, pour avoir un numérotage des lignes, ouvrir :

```
vim /etc/vim/vimrc
```

et décommenter les lignes (enlever les " devant !) ou les écrire :

set number	" Numérotation des lignes
syntax on	" Coloration syntaxique... A vous de voir
set cursorline choisir blue ou autres... highlight CursorLine ctermbg=grey	" Cela met la ligne courante en gris... Vous pouvez
set background& du précédent !)	" Devine tout seul la couleur du fond (complément

Vous pouvez chacun proposer vos écritures d'amélioration dans ce fichier vimrc... A vous lire !

Le site visité pour ce tuto :

<http://linux-attitude.fr/post/Vimrc-collaboratif>

Exemple de vimrc

On édite le fichier : ~/.vimrc

```
nano ~/.vimrc
```

Liste des raccourcis clavier pour la correction orthographique :

- z= sur un mot souligné affiche une liste de correction possible.
- zg rajoute un mot dans le dictionnaire
- zug pour annuler l'ajout au dictionnaire
-]s pour aller au prochain mot mal orthographié
- [s pour le précédent

Lien de vimrc configuré grave :

- [vimrc](#)

Conclusion

Ne sous-estimez pas ce logiciel (ni "vous-mêmes" chers debianeux adorés....) c'est tellement plusssssssssssss mieux pour vivre ensuite avec vot' pc... Yep !

Liens utiles

- <http://www.linux-france.org/article/appli/vi/vim/>
- http://www.yolinux.com/TUTORIALS/LinuxTutorialAdvanced_vi.html

1)

N'hésitez pas à y faire part de vos remarques, succès, améliorations ou échecs !

2)

sauf avec vim-tiny, mais qui voudrait n'installer que le vim-tiny, franchement 😊

3)

La commande qui l'indique :

```
dpkg -L vim-runtime |grep vimtutor
```

4)

tapez avant tout sur Echap

From:

<http://debian-facile.org/> - **Documentation - Wiki**

Permanent link:

<http://debian-facile.org/doc:editeurs:vim:vim>



Last update: **16/06/2020 07:47**