

L'éditeur VIM

Table des matières

| | |
|--------------------------------------|---|
| Introduction..... | 1 |
| Exemple de commandes : | 1 |
| Opérations sur les fichiers..... | 1 |
| Opérations sur les caractères..... | 2 |
| Opérations sur les mots..... | 2 |
| Opérations sur les lignes..... | 2 |
| Se positionner dans le fichier..... | 3 |
| Recherche..... | 3 |
| Recherche et remplacement..... | 3 |
| Opérations répétitives (macros)..... | 3 |
| Programmation..... | 4 |
| Drapeaux VIM..... | 4 |
| Shell..... | 4 |
| Conclusion..... | 4 |

Introduction

VIM est, avec **Emacs**, un des éditeurs de base les plus utilisés sous GNU/Linux.

Sa logique repose sur deux modes :

- ➔ un mode **insertion** dans lequel vous pouvez écrire votre texte
- ➔ un mode **commande (ESC)** qui utilise des raccourcis pour faire des actions

La plupart des raccourcis sont pensés « à l'anglaise », ce qui simplifie leur mémorisation.

À noter que dans les recherches/remplacements de texte, **VIM** utilise la puissante syntaxe des expressions régulières PERL (**regexp**), que l'on trouve communément sous GNU/Linux.

Exemple de commandes :

```
. : répéter la dernière commande
u (undo) : annuler le dernier changement
CTRL-r (redo) : rétablir le dernier changement
v (visual) : activation du mode visuel pour
sélectionner plusieurs lignes avec les flèches
gauche/droite/haut/bas, pour faire ensuite une
action sur ces lignes (copier, couper, joindre,
etc)
```

Opérations sur les fichiers

```
:r fichier : lire un fichier et l'insérer à
partir de la ligne suivante
:n (next) : aller au fichier suivant si
plusieurs fichiers ont été spécifiés lors de
l'ouverture sur la ligne de commande
:N : aller au fichier précédent si plusieurs
fichiers ont été spécifiés lors de l'ouverture
sur la ligne de commande
:w (write) : enregistrer
```

:q (**quit**) : quitter si aucune modification n'a été apportée
:q! (**quit**) : forcer à quitter en oubliant les modifications apportées
:wq ou **:x** : enregistrer et quitter
:wq! : enregistrer en écrasant le contenu et quitter

Opérations sur les caractères

i (**insert**) : insérer à la position courante
a (**append**) : insérer après le caractère courant
I (**insert**) : insérer au début de la ligne
A (**append**) : insérer à la fin de la ligne
r (**replace**) : remplacer le caractère courant
R (**replace**) : remplacer les caractères depuis le caractère courant
x : effacer le caractère courant
X : effacer le caractère à gauche du caractère courant

Opérations sur les mots

b (**begin**) : aller au mot précédent
B (**begin**) : aller au mot précédent en utilisant l'espace comme délimiteur
w (**word**) : aller au mot suivant
W (**word**) : aller au mot suivant en utilisant l'espace comme délimiteur

e (**end**) : aller à la fin d'un mot
E (**end**) : aller à la fin d'un mot en utilisant l'espace comme délimiteur
cw (**change word**) : changer le mot depuis la position courante
dw (**delete word**) : effacer un mot vers la droite (dont le mot courant)
3dw (**delete word**) : effacer 3 mots vers la droite (dont le mot courant)
db : effacer un mot vers la gauche (dont le mot courant)
3db : effacer 3 mots vers la gauche (dont le mot courant)

Opérations sur les lignes

C (**clear**) : effacer depuis le caractère courant jusqu'à la fin de la ligne
cc (**clear**) : effacer la ligne courante et se placer en mode insertion
o : insérer une nouvelle ligne après la ligne courante
O : insérer une nouvelle ligne avant la ligne courante
^ : aller en début de ligne
\$: aller en fin de ligne
yy ou **Y** : copier la ligne courante
3yy ou **3Y** : copier 3 lignes dont la ligne courante

p : coller le contenu du buffer après la ligne courante
P : coller le contenu du buffer avant la ligne courante
dd : effacer la ligne courante
3dd : effacer 3 lignes (dont la ligne courante)
:3,8d : effacer de la ligne 3 à la 8
J (**join**) : joindre la ligne courante et la suivante
3J (**join**) : joindre 3 lignes (dont la courante)

?toto : rechercher le mot toto depuis la position courante vers la début du fichier
n : aller au résultat de recherche suivant
N : aller au résultat de recherche précédent
:3,10g/toto : cherche le mot toto des lignes 3 à 10
***** : sur un mot, chercher la prochaine occurrence de ce mot
: sur un mot, chercher la précédente occurrence de ce mot

Se positionner dans le fichier

:1 ou **1G** ou **gg** : aller au début du fichier
:\$ ou **G** : aller à la fin du fichier
:19 : aller à la ligne 19
(: aller au début d'une phrase
) : aller à la fin d'une phrase
{ : aller au début d'un paragraphe
} : aller à la fin d'un paragraphe
H : se positionner en haut de l'écran
M : se positionner au milieu de l'écran
L : se positionner en bas de l'écran

Recherche

/toto : rechercher le mot toto depuis la position courante vers la fin du fichier

Recherche et remplacement

:s/toto/titi/g : remplace tous les *toto* par des *titi* (**g**=changement global)

Opérations répétitives (macros)

C'est un des points dans les traitements par lot : VIM permet d'enregistrer des opérations dans un buffer, puis de réappliquer ce buffer à tous les fichiers ouverts.

qa : enregistrer une séquence d'action dans le buffer **a** - il faut faire **ESC** pour arrêter l'enregistrement
@a : rejouer la séquence du buffer **a**
3@a : rejouer 3x la séquence du buffer **a**

Programmation

```
% : rechercher l'accolade appairée à l'accolade
courante (pour trouver où commence et où finit
une fonction)
= : réindenter le code source sélectionné
gg=G : réindenter tout le fichier
```

Drapeaux VIM

Les options de VIM se comptent en centaines et il est inutile de vouloir toutes les apprendre par cœur. L'idée générale est que chaque option contrôle un comportement de l'éditeur. Si vous voulez un (gros) aperçu des possibilités offertes - cf : <http://david.blanchet.free.fr/vim/doc/fr62017/options.txt.html>

```
:set ic : activer la casse pour les recherches
:set noic : désactiver la casse pour les
recherches
:set nu : activer la numérotation des lignes
:set nonu : désactiver la numérotation des
lignes
```

Shell

```
:!bash : ouvrir une nouvelle console en gardant
l'éditeur VIM actif. Pour revenir dans
l'éditeur, il suffit de fermer la console avec
exit
```

Conclusion

Ce petit tutoriel est loin d'être exhaustif : il ne fait que proposer mes raccourcis les plus « courants » à l'usage, qui ne seront peut-être pas les vôtres...

La personnalisation du fichier `~/.vimrc` permet d'imposer de très nombreuses options, comme la longueur et le type des tabulations, indiquer si l'on veut des espaces ou des tabulations quand on appuie sur la touche `TAB`, les couleurs d'avant plan et de fond, les indentations variables suivant le type de fichier, etc.

Les administrateurs peuvent aussi imposer au départ les réglages de **VIM** pour tous les usagers, via le fichier `/usr/share/vim/vimrc`.

VIM est ainsi un véritable « couteau suisse » extensible à souhaits et particulièrement puissant.

Il est par exemple redoutable pour nettoyer des fichiers texte avant introduction dans des bases de données SQL, via les expressions régulières PERL et autres outils courants de GNU/Linux (awk, sed, grep, etc) !

Considérez donc son (long) apprentissage comme un investissement sur le long terme, et n'essayez pas d'absorber tout d'un coup. Plus vous travaillerez, plus vos raccourcis viendront naturellement sous le clavier !

Bon apprentissage à vous !