

L'éditeur vi

1. Présentation

L'éditeur Unix par défaut se nomme **vi** (*visual editor*). S'il n'est pas des plus ergonomiques par rapport à des éditeurs en mode graphique, il a l'avantage d'être disponible et d'utiliser la même syntaxe de base sur tous les Unix. Chaque Unix propose généralement une syntaxe étendue au-delà de la syntaxe de base. L'éditeur vi sous Linux se nomme vim. Vim respecte toute la syntaxe de vi, la réciproque n'étant pas vraie. Vi est petit : il occupe peu d'espace disque, consomme peu de mémoire.

```
vi [options] Fichier [Fichier2 ...]
```

Vi n'a pas de menus, pas d'interface graphique et n'est pas intuitif. Cela nécessite de connaître par cœur un certain nombre de raccourcis-claviers pour pouvoir l'utiliser. Si l'apprentissage est un peu difficile, une fois maîtrisé vi se révèle rapide et pratique, au point qu'on va plus vite qu'avec des éditeurs de texte graphiques.



Le débat opposant les partisans de emacs (ou d'autres éditeurs) et ceux de vi n'a pas lieu d'être. Tout système Linux (et Unix) dispose quoi qu'il arrive de l'éditeur vi, le rendant incontournable. Si vous en avez la possibilité, vous pouvez par la suite installer l'éditeur qui vous plaira. Ce ne sera pas toujours possible, notamment en entreprise, sur des serveurs, etc.

2. Fonctionnement

Il y a trois modes de fonctionnement :

- mode **commande** : les saisies représentent des commandes. On y accède en appuyant sur [Echap]. Chaque touche ou combinaison de touches déclenche une action (suppression de lignes, insertions, déplacement, copier, coller, etc.).
- mode **saisie** : c'est la saisie de texte classique.
- mode **ligne de commande** : une ligne en bas d'écran permet de saisir des commandes spéciales, validée avec la touche [Entrée]. On y accède en appuyant, en mode commande, sur la touche « : ».

Quand vous lancez vi, il est par défaut en mode commande. Pour commencer à taper du texte, ce qui est vite énervant quand on ne connaît pas vi, il faut taper une commande d'ajout ou d'insertion : a ou i. Donc la commande [Echap] i permet de saisir du texte. Pour quitter, vous pouvez passer par le mode ligne de commande. Vous tapez [Echap] : et enfin q et [Entrée]. C'est simple !



Si la commande saisie en mode commande n'est pas une commande de saisie, vous ne quittez pas ce mode, et vous n'êtes pas obligé de taper [Echap] avant de saisir une nouvelle commande. Par exemple le « x » supprime un caractère. Si vous êtes déjà en mode commande, vous pouvez appuyer dix fois sur « x » sans appuyer sur [Echap] entre chaque appui.

Il y a bien entendu une pointe d'ironie. Il n'est pas évident de connaître tous les raccourcis, l'auteur de ces lignes lui-même ne les retient pas tous. Aussi, outre l'existence de guides en accordéon dans la presse spécialisée, voici la liste des commandes les plus utilisées sous forme de tableaux. Quand la commande est précédée du « : », c'est une commande à saisir en ligne de commande.

Si vous êtes perdu, dans tous les cas et quoi qu'il arrive, un appui sur la touche [Echap] revient toujours en mode commande.

3. Les commandes

a. La saisie

Les actions suivantes sont à effectuer en mode commande. Elles doivent être précédées d'un appui sur [Echap] : [Echap] a, [Echap] i, etc.

Commande	Action
a	Ajout après le caractère actuel.
A	Ajout de texte en fin de ligne.
i	Insertion devant le caractère actuel, comme dans un traitement de texte.
I	Insertion de texte en début de ligne.
o	Ajout d'une ligne sous la ligne actuelle.
O	Insertion d'une ligne au-dessus de la ligne actuelle.

b. Quitter et sauver

Pour mémoire, les « : » signifient que la commande se tape en ligne de commande : [Echap] :, saisie de la commande, puis [Entrée].

Commande	Action
zz	Sauve le fichier et quitte.
:q!	Quitte sans sauver.
:q	Quitte si le fichier n'a pas été modifié (apparition d'un message d'erreur sinon).
:w	Sauve le fichier. Vous pouvez préciser un nom à la suite.
:wq ou :x	Sauve et quitte.
1,10w fic	Sauve les lignes de 1 à 10 dans fic.

c. Déplacement

Vous allez probablement sourire, mais il existe encore des claviers sans touches directionnelles, et des logiciels et des interpréteurs de commande ne sachant pas les interpréter (exemple du shell ksh mal configuré sur certains Unix). Sur ces machines, c'est le mode de saisie de vi qui est activé par défaut, et même sous vi il est possible de se passer des touches fléchées, en mode commande.

Commande	Action
h	Aller vers la gauche.
l (petit L)	Aller vers la droite.
k	Aller vers le haut.
j	Aller vers le bas.

Commande	Action
O (zéro)	Début de ligne.
:O	Début de fichier (première ligne).
\$	Fin de ligne.
:\$	Fin de fichier (dernière ligne).
w	Aller au mot suivant.
b	Aller au mot précédent.
f<c>	Sauter au caractère <c> suivant.
Control+f	Avance d'un écran (Forward).
Control+b	Reculé d'un écran (Backward).
G	Dernière ligne du fichier.
<n>G	Saute à la ligne « n » (ex : 10G va à la 10 ^{ème} ligne).
:<n>	Idem (:10 va à la 10 ^{ème} ligne).

d. La correction

Commande	Action
x	Efface le caractère sous le curseur.
X	Efface le caractère devant le curseur.
r<c>	Remplace le caractère sous le curseur par le caractère <c>.
dw	Efface depuis le curseur jusqu'à la fin du mot.
d\$ ou D	Efface depuis le curseur jusqu'à la fin de la ligne.
dO	Efface depuis le début de la ligne jusqu'au curseur.
df<c>	Efface tout jusqu'au caractère <c>.
dG	Efface tout jusqu'à la dernière ligne, y compris la ligne actuelle.
d1G	Efface tout jusqu'à la première ligne, y compris la ligne actuelle.
dd	Efface la ligne actuelle.
u	Undo. Annule la dernière action.

Ces commandes peuvent être répétées. 5dd supprime 5 lignes, 4dw 4 mots, 5x 5 caractères, etc.

e. Recherche dans le texte

Contrairement à un éditeur de texte classique, vi peut rechercher autre chose que des mots simples et fonctionne à l'aide de caractères spéciaux et de critères. La commande de recherche est le caractère /. La recherche démarre du caractère courant jusqu'à la fin du fichier. Le caractère ? effectue la recherche en sens inverse. On indique ensuite le critère, puis [Entrée].

/echo

recherche la chaîne 'echo' dans la suite du fichier. Quand la chaîne est trouvée, le curseur s'arrête sur le premier caractère de cette chaîne.

La commande n permet de continuer la recherche dans le sens indiqué au début. La commande N effectue la recherche en sens inverse.

Quelques critères

- /[FfBb]oule : Foule, foule, Boule, boule.
- /[A-Z]e : tout ce qui commence par une majuscule avec un e en deuxième position.
- /[A-Za-Z0-9] : tout ce qui commence par une majuscule, une minuscule ou un chiffre.
- /^[^a-z] : plage négative : tout ce qui ne commence pas par une minuscule.
- /vé.o : le point remplace un caractère, vélo, véto, véro, ...
- /Au*o : l'étoile est un caractère de répétition, de 0 à n caractères, Auo, Auto, Automoto, ...
- /. * : l'étoile devant le point, une chaîne quelconque de taille variable, le "." représentant un caractère.
- /[A-Z][A-Z]* : répétition du motif entre [] de 0 à n fois, recherche d'un mot comportant au moins une majuscule (en début de mot).
- /^Auto : le ^ indique que la chaîne recherchée devra être en début de ligne.
- /Auto\$: le \$ indique que la chaîne recherchée devra être en fin de ligne.

f. Commandes de remplacement

Pour remplacer du texte, il faut se placer au début de la chaîne à modifier, puis taper l'une des commandes suivantes. Ensuite tapez simplement votre texte. Le principe est simple, il est identique à celui de la suppression : le morceau précisé est supprimé et vi passe en mode d'insertion pour la saisie.

Commande	Action
cw	Remplace le mot courant.
c\$	Remplace jusqu'à la fin de la ligne.
c0 (zéro)	Remplace jusqu'au début de la ligne.
cf<x>	Remplace jusqu'au prochain caractère <x>.
c/<rech>	Remplace jusqu'à la prochaine occurrence de la chaîne <rech>.

g. Copier-Coller

La commande v permet une sélection visuelle. Le texte est surligné et vous pouvez déplacer le curseur pour sélectionner le texte. Utilisez ensuite l'une des commandes suivantes :

- pour couper (déplacer), c'est la commande « d » ;

- le c fait presque la même chose, mais vi reste en mode d'édition ;
- pour coller le texte à l'endroit choisi, c'est la commande p (derrière le caractère) ou P (devant le caractère). Si c'est une ligne complète qui a été copiée, elle sera placée en dessous de la ligne active.

Les actions suivantes sont possibles en mode commande :

- Pour copier une ligne : yy.
- Pour copier cinq lignes : 5yy.
- Pour placer les lignes copiées à un endroit donné : p.
- L'éditeur vi dispose de 26 tampons pour y stocker les données que l'on peut nommer comme on le souhaite. On utilise pour cela le ".
- Pour copier cinq mots dans la mémoire m1 : "m1y5w.
- Pour coller le contenu de la mémoire m1 à un endroit donnée : "m1p.

h. Substitution

La substitution permet de remplacer automatiquement plusieurs occurrences par une autre chaîne.

```
: [1ere ligne, dernière ligne] s /Modèle/Remplacement / [gc]
```

Les numéros de lignes sont optionnels. Dans ce cas la substitution ne se fait que sur la ligne courante. En remplacement des numéros de lignes, . détermine la ligne courante, 1 la première ligne, \$ la dernière ligne.

Le modèle est un critère de recherche présenté dans ces dernières pages. Remplacement est une chaîne quelconque qui remplacera le modèle.

Par défaut seule la première occurrence est remplacée. La lettre g indique qu'il faut remplacer toutes les occurrences. Avec c, vi demande une confirmation pour chacune des occurrences. L'exemple remplace toutes les occurrences de Unix ou Unix en UNIX.

```
:1,$s/[Uu]nix/UNIX/g
```

i. Autres

Édition avancée

Voici quelques commandes pratiques.

<code>:r fic</code>	Insère le contenu de fic à partir de l'endroit actuel.
<code>:! cmd</code>	Exécute la commande. Appuyez sur [Entrée] pour revenir sous vi.
<code>:r! cmd</code>	Le résultat de la commande est inséré à l'endroit actuel.
<code>:e fic</code>	Charge le fichier fic pour édition.
<code>:e#</code>	Commute entre les divers fichiers ouverts.

Commande set

La commande **set** permet de configurer l'éditeur et d'accéder à ses options :

- set all : affiche l'ensemble des options possibles.
- set number (ou nu) / nonumber (ou nonu) : affiche/supprime les numéros de lignes.
- set autoindent / noautoindent : l'indentation est conservée lors d'un retour à la ligne.

- `set showmatch / noshowmatch` : lors de la saisie d'une accolade ou d'une parenthèse de fermeture, celle d'ouverture est affichée un très court instant, puis l'éditeur revient au caractère courant.
- `set showmode / noshowmode` : vi affichera une ligne d'état (INPUT MODE).
- `set tabstop=x` : définit le nombre de caractères pour une tabulation.

Vim propose bien souvent la coloration syntaxique. Il détecte le type de fichier chargé et colore les lignes, les mots clés, etc. Vous pouvez désactiver la coloration :

- `syntax off` pour désactiver ;
- `syntax on` pour activer.