

Guide de survie sous VI ou VI par l'exemple.

Charles VIDAL

vidalc@univ-mlv.fr

-
1. [VI KESAKO](#)
 2. [Pourquoi Apprendre VI](#)
 3. [L'utilisation et les pieges a eviter de VI.](#)
 1. [VI a deux modes :](#)
 4. [Premier pas ...](#)
 1. [voici quelques commandes](#)
 2. [Exemples d'utilisation:](#)
 5. [copier-coller](#)
 6. [substitution, ou VI puissance 10:](#)
 7. [Action contexte sensitif, ou substitution puissance 10](#)
 8. [visualisation et recherche](#)
 9. [Changement du Mappage du clavier](#)
 10. [Entree Sortie sous VI.](#)
 11. [AU SECOURS](#)
 12. [DIVERS.](#)
 13. [Truc pour les codeurs.](#)
 1. [Autres trucs:](#)
 2. [Macro pour le C:](#)
 3. [utilisation de ctags .](#)
 14. [Les options](#)
 1. [Options Booleennes:](#)
 2. [Autres Options:](#)
 15. [Bidouille plus complexe.](#)
 16. [le fichier de ressource.](#)

VI KESAKO ???

Vi est un editeur visuel (prononce Via), a l'oppose des editeurs non visuels, par exemple sed,ex,ed,edlin. Vi a ete ecrit par William Joy comme une part de la distribution BSD d'UNIX . Puis At&T l'a utilise, et maintenant c'est devenu un standard sous Unix. Cette standardisation de l'editeur VI permet donc de retrouver des versions de VI sur tout les systemes actuels : Unix, AmigaOS, ST-OS, DOS, OS-2, SYSTEM 7, WIN3.1, WIN95, et bien d'autre future(BE-OS).

Ceci dit les versions de VI varient un peu, certaines commandes ne marchent pas forcement de la meme maniere partout.

Pour apporter de l'eau a mon moulin, j'ai utilise VI sur un terminal X, un terminal vt100, Win95, DOS, et meme un minitel (c'est dire).

Pourquoi apprendre VI ?

C'est vrai, VI est penible a apprendre, surtout aujourd'hui ou les nouveaux editeurs sont tous muni d'un minimum d'interface graphique. Mais la puissance de VI surpasse de beaucoup d'autres editeurs plus conviviaux. On peut quasiment tout faire sous VI, la difficulte est de savoir comment VI peut le faire.

Je suis un peu proselytiste a propos de VI (je pourrais l'etre a propos d'Emacs, mais la n'est pas le propos), et voyant un ami utilisant un editeur graphique (MOTIF/OSF), gourmand en ressource, je lui dis qu'il devrait utiliser VI . Ajoutant que j'utilisais VI sous DOS, il me rit au nez, traite VI des pires maux, et ose meme le comparer a edit (editeur du DOS).

Apres 10 minutes, lui montrant une petite partie des possibilitees de VI, fonctionnalitees impossibles sous edit, et meme sous la plupart des editeurs DOS meme borland, il abdique, et me demande ou se procurer la version DOS.

J'espere vous aider a apprehender VI, et vous faire votre propre idee.

L'utilisation et les pieges a eviter de VI .

Attention, au premier contact de VI, il est probable que vous soyez deconcerte, totalement perdu.

Il y a certaines choses a connaitre.

L'interface avec l'utilisateur se fait uniquement par le biais du clavier. Donc, ne vous enervez pas sur celui-ci, la situation ne fera qu'empirer .

la touche DEL n'efface pas les caracteres, ca trouble beaucoup au debut.

VI a deux modes :

1. un mode **INSERT** on tape 'v', un 'v' s'insere dans le texte .
2. un mode **COMMAND** on tape ':q', on quitte vi .

Toute la difficultee du debutant est de savoir dans quel mode il est. En cas desespere, appuyez sur ESC puis tapez ":se smd", ou ":se showmode". Cette commande, montre en bas a droit, le mode ou l'on se trouve.

Au lancement de VI, on est dans le mode **COMMAND** , par default.

Pour passer du mode **COMMAND** au mode **INSERT** , il y a plein de touches, mais la plus utilisee est 'i' pour mode 'INSERT' ! Pour passer du mode **INSERT** au mode **COMMAND** , ici une seule touche ESC (ou CTRL-C). Une fois en mode **COMMAND** , on peut executer une commande de differente facon.

Exemples:

- dd supprime la ligne courante.
- 5dd supprime 5 lignes a partir de la ligne courante.
- :3,5dd supprime la ligne 3 a la ligne 5 .
-

Apres une commandes commençant par ":" ou "", appuyer sur ENTER pour l'executer . Beaucoup de commandes peuvent s'executer de la meme facon.

Il y a des exeptions : *Par Exemple w* vous positionne sur le prochain mot, alors que ":w" sauve le fichier.

Attends, mais si j'ai pas de touche ESC sur mon clavier, comment utiliser VI?

essaie CTRL-[, CTRL-3 ,ou CTRL-C, sous DEC terminal, F11 est utilise comme touche ESC.

Premier pas ...

lancez le programme vi (elvis ou autre clone).

l'ecran s'efface .

appuyez sur 'i' (pour passer en mode **INSERT**), si en bas de l'ecran s'affiche insert, sautez la ligne suivante. Sinon apuyer sur ESC puis tapez ":se showmode" ENTER puis sur i. Frapper x, un "x" apparait a la premiere ligne (sinon recommencer depuis debut) frapper ensuite ESC (maintenant vous etes en mode **COMMAND**), puis x, le "x", qui a ete precedement inserer, disparaît. Vous devez comprendre, maintenant, la difference entre le mode **COMMAND** et le mode **INSERT** .

En mode **INSERT** , x insere le caractere "x".

En mode **COMMAND** , x, qui n'est plus le caractere mais une commande, supprime le caractere courant.

Pour finir, etant toujours en mode **COMMAND** frapper ":q!" .Voila vous etes sorti de VI

BRAVO

Faites la meme operation mais a la fin taper ":w" toto puis ":q", le fichier toto sera alors cree

.

voici quelques commandes

Commandes avec passage en mode **INSERT**

a	ajoute apres la lettre courante.
A	ajoute apres la ligne courante.
o	ajoute une ligne en dessous
I	insere en debut de ligne.
cw	change le mot sur lequel on se trouve

Commande sans passage en mode **INSERT**

x	enleve le caractere a la position courante
:10	la ligne courante devient la ligne 10
dw	supprime le mot sur lequel on se trouve position courante.
:q	quitte
:w	sauve
:q!	quitte sans sauver.

Cette liste n'est pas exhaustive, c'est un minimum pour debuter. D'autres commandes seront donnees dans la suite du document. Mais au debut, ne soyez pas trop ambitieux, maitrisez les comandes ci-dessus, vous serez deja tres bon.

Exemples d'utilisation:

envoi:

vi edite un nouveau fichier sans nom.

vi toto.c edite le fichier toto.c, s'il n'existe pas il sera cree a la prochain sauvegarde.

vi *.txt edite tous les fichiers *.txt pour passer au suivant faire en mode commande :n et :N pour passer au precedent.

remarque Un grand avantage de VI est qu'il est tres rapide a s'executer. Sur HP-UX, 1 minute pour editer 15 Mega.

copier-coller

<

:10,20t 100 copie les lignes de 10 a 20 a la 100 eme ligne .

:10,20m 100 deplace les lignes de 10 a 20 a la 100 eme ligne .

:10,20y met dans le buffer les 10 lignes de 10 a 20.

:100,\$m 50 deplace les lignes 100 jusqu'a la fin du fichier vers la 50 eme ligne.

yy copie la ligne courante dans le buffer .

p colle le buffer dans le texte.

remarque

le couper-coller marche par la commande 'dd' apres on deplace le curseur puis 'p' (comme paste) coller la ligne couper; de meme avec "10x" pour les caracteres.

substitution, ou VI puissance 10:

:%s/window/linux/g	change windows par linux dans tout le fichier.
:%s/windows/linux	change windows par linux dans tout le fichier, mais une seule fois par ligne
:%s/windows/linux/gc	change windows par linux dans tout le fichier, mais avec confirmation.
:10,20s/windows/linux/g	change windows par linux dans les lignes de 10 a 20.
	met en majuscule la ligne courante.
:s/.*/U&/	U majuscule L minuscule
:%s/ `^ m//	equivalent de la commande dos2unix pour avoir `^m appuyer sur CTRL-V puis m enter .
:s/[a-zA-Z]/1&/g	ajoute apres tout caractere dans [a-zA-Z] un 1 devant.
	elimine windows en debut de ligne.
:%s `^ windows//	^ veux dire debut de ligne \$ veux dire fin de ligne
:/windows/-2,+4d	elimine 5 lignes autour de la premiere occurence de "windows"
:s/[ABC]/[abc]/g	A devient a, B devient b, C devient c .

Action contexte sensitif, ou substitution puissance 10

On peut demander a VI de faire une action sur une phrase uniquement si cette derniere contient un motif textuel.

Syntaxe: ":g/chaine/commande"

:g/micrisoft/s/windows/linux/g	changes windows par linux uniquement dans les lignes ou "micrisoft" se trouve .
:g/ `^[]*\$d	supprime toutes les lignes vides.

:g/linux/t 0	copie en haut du texte toutes les lignes contenant linux
:g/linux/cd 80	centre tous les lignes contenant linux
:g!/linux/d	efface toutes les lignes ne contenant pas linux
:v/linux/d	idem

visualisation et recherche

:/linux/-2,+4p	affiche 5 lignes autour de la premiere occurence de "linux".
^linux	recherche linux en debut de ligne
:10,20nu	visualisation des lignes 10 a 20 avec affichage des numeros de lignes.
/lin[uU]?x/	recherche linux ou linUx
n	Recherche la nouvelle occurence.

Changement du Mappage du clavier

les touches des claviers peuvent acquerir une fonctionalitee, en mode **COMMAND** .

'v' qui n'en n'a pas, on peut lui en affecter une:

```
:map v 10dd
```

Après la touche 'v' efface 10 lignes a partir de la position courante, en mode **COMMAND** .

```
:map! v 10dd
```

v aura la fonctionnalitee dans les 2 modes.

Exemple:

```
:map! ^vDELETE ^vESC\xi
```

Maintenant, la touche DELETE supprime dans les 2 modes

de meme:

```
:map! ^vBACKSPACE ^vESC\xh
```

Maintenant, la touche BACKSPACE marche aussi .

Entree Sortie sous VI.

...	
:w	saue le fichier edite.
:e	open le meme fichier que celui edite, si on n'as pas saue, c'est un puissant undo
:r toto.c	insere le fichier toto.c a la position courrante.
:10,20w toto	saue les lignes de 10 a 20 dans le fichier toto
:10,20w toto	ajoute au fichier tot les ligne 10 a 20 du fichier edite
:r!ls -l	insere le resultat de la commande ls -l dans l'editeur a la position courante

AU SECOURS

```
:vi    permet de retablir l'affichage
:q!    permet de sortir sans rien changer.
:e!    permet de reediter le fichier sans sauver une connerie ( ca arve a tous le monde ).
:u     undo annule la derniere insertion.
:se all affiche toutes les options.
```

DIVERS

l'appuis de la touche '.' en mode COMMAND repete soit la derniere insertion ou la derniere command, ceci est hyper pratique .

- la touche \$ envoie a la fin de ligne.
- la touche ^ envoie au debut de ligne.
- la touche w envoie au debut mot suivant.

..	
r	remplace la lettre courante
cw	change le mot courant.
2cc	change 2 lignes a partir de la ligne courante. pour ces denieres commandes il y a un passage en mode INSERT .
:10ce 80	centre la ligne 10 sur 80 colonnes.
:10,20ce 80	centre les lignes de 10 a 20 sur 80 colonnes.
:right	aligne a droite la ligne courante
:left	aligne a gauche la ligne courante.
:10,20 >1	insertion d'une tabulation pour les lignes de 10 a 20 .
	<1 en enleve une
:10,20j	joint toutes les lignes de 10 a 20 en une seule
:map #7 dd	associe la touche F7 a la commande dd
:color light cyan on blue	cyan sur fond bleu
10,20w tmp	sauve les lignes de 10 a 20 dans le fichier tmp.
:ab GNU gnu is not unix	abreviation GNU devient gnu is not unix attention a la recursivite (GNU - gnu et non GNU - GNU)
:ab email vidalc@mlv-univ.fr	pratique pour son mail
:ab	affiche toutes les abreviations
:unab email	elimine l'abreviation email.
:10,20!sort	remplace le texte (10 -> 20) part la sortie du philtre unix sort dont l'entree a ete ce texte
:so .exrc	execute toute les commandes VI du fichier .exrc
ma	deplace le curseur puis(marque la position courante par a)
mb	puis (marque la position courante par b)
'a	puis (va a la marque a)
'b	puis (va a la marque b)
:jumps	puis (affiche les sauts)
:marks	affiche toutes les marques du texte

Truc pour les codeurs.

Il faut savoir que VI a ete fait pour les programmeurs UNIX et il est donc normal de retrouver des fonctionnalitees simplifiant la programmation.

Un langage du premier age, LISP, (LANGUAGE OF IDIOT AND STUPID PARENTHESES) est un enfer de parentheses.

Heureuseusement il y a VI.

Si dans un programme LISP, vous tapez '%' en mode **COMMAND** (c'est la derniere fois que je le precise), VI vous montrera la premiere parenthese ouvrante trouvee, puis de nouveau '%' et hop VI montre la parenthese fermante corresconpande.

Aucun editeur sous DOS commercial, fait ca!!! Meme pas les editeurs BORLAND. C'est pas dur pourtant..., et tellement pratique.

la commande '%' fonctionne avec '{','}','(',')','[',']', enfin grosomodo tout les caracteres qui s'ouvrent et se ferment.

Ca marche aussi, avec le C ou C++, ou autre.

Donc pas de probleme pour verifier les accolandes, sauf quand elles se trouvent dans une chaine de caracteres (dommage, probleme resolu sous Emacs).

Autres trucs:

-
- la commande) avance sur la prochaine phrase.
- la commande } avance sur le prochaine paragraphe.
- la commande]] avance sur la prochaine section.

Ceci permet de saute de fonctions en fonctions en C ou C++, entre autre. Meme commandes avec les caracteres ouverts, mais le deplacement ce fait dans l'autre sens.

Macro pour le C:

Pour obtenir le ^ [appuyer sur CTRL-V puis CTRL-ESC.

Ce caractere `^` [est la simulation ESC, donc passage en mode **COMMAND** .

- :map `^ A o{ `^ [o} `^ [O & CTRL-A insert deux accolade et vous positionne entre les accolades en mode **INSERT** .
- :ab defoo `^ [:O `^ MO#define & abreviation defoo insert #define en haut du fichier
- :ab foo for (){ `^ [o} `^ [k\$5hi & abreviation foo insert for avec parenthese, et tout et tout..
-

utilisation de ctags .

Allez dans le repertoire ou il y a vos sources sous un shell quelconque, tapez "ctags *.c" (creation d'un fichier tags). Puis sous VI , dans un programme source tapez en mode **COMMAND**:

```
":ta nom d'un fonction"
```

Et hop vous voila ammene dans le fichier contenant la dite fonction , a l'endroit ou elle se trouve .

Les options

Il existe une commande sous VI, qui n'influe pas sur le texte edite, mais sur l'editeur VI lui-meme.

Cette commande est ":set" ou ":se" comme ci dessus, il y a plein de raccourci clavier.

la syntaxe de cette commande est

- :se affiche les options modifies.
- :se all affiche toutes les options possibles
- :se option positionne l'option Booleenne a Vrai (True)
- :se no option positionne l'option Booleenne a Faux (False)
- :se option =valeur positionne l'option a la valeur valeur.
- :se option ? affiche la valeur de l'option.

voici quelques options tres utile ou loufoque.

Options Booleennes:

- :se autoindent
- :se ai apres ENTER la ligne suivante commencera a la meme indentation.
- :se binary
- :se bin permet d'editer des fichiers binaire (decidement VI peut tout faire) .
- :se ignorecase A la recherche de chaine, VI ignore minuscule majuscule. et au substitution.
- :se ic A la recherche de chaine, VI ignore minuscule majuscule et au substitution.
- :se number
- :se nu affiche le numero de lignes a gauche. (tres pratique)
- :se visualbell
- :se vb inverse le BEEP par un effet visual.
- :se readonly
- :se ro ecrit un fichier meme si c'est proteger en ecriture
- :se ruler
- :se ru affiche les coordonnees du curseur (pas sur HP-UX)
- :se lisp montre automatiquement la coorespondance des caracteres ouvert et fermer .(marche pas sous elvis)
- :se revins
- :se ri inversion des caracteres de droite a gauche (Il peut vraiment tout faire meme les truc fous)
- :se showmode
- :se smd indique s'il on est en mode INSERT ou COMMAND .
-

Autres Options:

- :se undolevels 100
- :se ul 100 positionne la possibilitee de faire 100 fois undo.
- :se history 20
- :se hi 20 positionne l'historique des commandes a 20.

Bidouille plus complexe.

yyp	Duplique une ligne.
y\$	Met dans le buffer, tout de la position courante a la fin de la ligne .
ddp	Bouge une ligne vers le bas .
xp	Deplace le caractere courant apres le caractere suivant.

:map ctrl-v<><>ctrl-g	essaye donc! Ca ralenti beaucoup VI .
:map ctrl-v<><>ctrl-g	idem
:map v Gdd"Pj	inexplicable.
d%	efface jusqu'a la prochaine parenthese fermante

le fichier de ressource.

A chaque lancement de VI, le fichier .exrc est lu

Mettez dans ce fichier:

se nu

se ruler

Pour avoir les options nu et ruler automatiquement, quand on lance VI.

On peut aussi mettre des touches de commande personnelle via map .

remarque : ruler ne marche pas partout (sous HP-UX il n'y a pas).

copyright charles vidal 1997 :)

merci a **pierre DITTGEN** pour son aide.

Toutes suggestions sont les bienvenues.