

vi : l'essentiel et plus !

Lancement

Vi [-r] fic1...

-r recover

en arguments plusieurs fichiers permis

Modes de fonctionnement

Mode commande : C'est le mode initial.

Il faut être en mode commande pour toute action qui n'est pas de la saisie (ex. : déplacement du curseur).

On y repasse en appuyant sur `escape`

Mode insertion : Mode réservé à la saisie.

On passe dans ce mode par les touches :

`a, i, c, o` `A, I, C, O`

Mode Ex : Les commandes en mode Ex commencent par : `:`

Ce sont des commandes : (1) de paramétrage de vi, (2) « à la ed » (substitution, manipulation de fichiers,...)

Généralités

On peut taper un facteur multiplicatif devant beaucoup de commandes **vi** (celles qui sont précédées de « * »)

L'unité par défaut des commandes à deux lettres identiques (ex :dd) est la ligne. On peut changer cette unité en un autre objet textuel (ex : w=word) en la passant en argument à la place de la 2^{ème} lettre du mnémonique (ex : dw, d\$, d0,d/chaine, 5d->, d'a, etc...).

Les expressions régulières (re) sont calquées sur celles de sed ou grep (ex :[0-9][0-9]*.*[^abc]\$)

Insertions

a	append	(derrière le curseur)
i	insert	(devant le curseur)
o	open	(en dessous de la ligne courante)
c	change	(voir cc dans les substitutions)
A	Append	(en fin de ligne courante)
I	Insert	(en début de ligne courante)
O	Open	(en dessus de la ligne courante)

Déplacements et objets textuels

*h	←	un caractère en arrière
*j	↓	une ligne en dessous
*k	↑	une ligne en dessus
*l	→	un caractère en avant
*w (W)	word	un mot en avant (début mot)
*e (E)	word	un mot en avant, en fin (end)
*b (B)	back word	un mot en arrière début de ligne
*\$		fin de ligne
'a		ligne marquée a (par : ma)
*/re		prochaine occurrence de re
*?re		précédente occurrence de re
n		après/re ou ?re pour répéter
N		= n, en inversant le sens

Déplacements purs

:v		va en ligne v
:l		va en début de fichier
:\$		va en fin de fichier
:: =		Affiche le n° de ligne courante
vG	Go	va en ligne v (dernière par défaut)
H	Home	va en haut à gauche de l'écran
M	Middle	va en milieu à gauche de l'écran
L	Last	va en bas à gauche de l'écran
*^D	down	descend d'un demi écran
*^U	Up	monte d'un demi écran
*^F	Forward	descend d'un écran
*^B	Backward	monte d'un écran

Destruction, mise en mémoire, restitution

*x	détruit un caractère
*dd	détruit une ligne (et la met en mémoire)
*yy	met en mémoire une ligne (c'est tout !)
p (min)	restitue derrière la ligne courante
P (MAJ)	restitue devant la ligne courante
*< φ yy	met l ligne en mémoire dans le buffer φ

φ est une lettre de l'alphabet. L'utilisation des buffers est possible avec les commandes dd, yy, p, P.

Substitutions

*r φ	remplace le caractère courant par φ
*cc	change une ligne (par n lignes)
<code>:s/re1/re2/[g]</code> substitution « à la ed »	

Manipulation de fichiers et fin de session

:w [fic]	sauve les modifications [sur fic]
:q	sort de vi, si les modifs ont été sauvées
:q !	sort de vi sans faire de sauvegarde
:x ou :wq	sort de vi en faisant la sauvegarde
ZZ	pareil à :x
:. r fic	insère fic après la ligne courante
:e fic	édite le fichier fic
:n	édite le fichier suivant (vi multi fichiers)
:f	affiche le nom du fichier courant
:. r fic	insère fic après la ligne courante

Commandes diverses

ma	marque la ligne avec a (lettre de a à z)
*J	joint 2 (ou v+1) lignes
*~	capitalisation
.	ré-exécute la commande précédente
u (U)	undo : défaire l' (les) action précédente
%	équilibre « (« avec ») » et « {« avec »} »
*>><<	indente (supprime l'indentation)
!<cde shell>	lance la commande shell
*!!	lance un filtre UNIX sur la ligne

Paramétrage

:set all	liste toutes les options
:set [no] nu	[dê]numérotage des lignes
:set smd	showmode (insertion/comde)
:map μ<macro>	(consulter le manuel!!!)