

L'éditeur V.I.



Table des matières

1 Éditeur de texte VI.....	2
1.1 Introduction.....	2
1.2 Fonctionnement.....	2
1.2.1 Lancement.....	2
1.2.2 Modes de travail.....	3
1.2.3 Cas pratique.....	3
1.3 Les commandes.....	5
1.3.1 Conventions d'écriture.....	5
1.3.2 Les commandes.....	5
1.3.2.1 Déplacement du curseur.....	5
1.3.2.2 Insertion.....	6
1.3.2.3 Recherche.....	6
1.3.2.4 Remplacement de texte.....	6
1.3.2.5 Effacement de texte.....	7
1.3.2.6 Transfert de Texte.....	7
1.3.2.7 Registres nommés.....	8
1.3.2.8 Commandes "ex" (ou de bas de page).....	8
1.3.2.9 Fin de session VI.....	9
1.3.3 Commandes diverses.....	9
1.4 Les variables.....	9
1.4.1 Commutateurs.....	10
1.5 Autres éditeurs.....	11
1.5.1 Vim.....	11
1.6 En conclusion.....	11

1 Éditeur de texte VI

1.1 Introduction

Le but essentiel de ce document est de présenter l'éditeur de texte le plus connu et le plus utilisé du monde Unix : l'éditeur VI (pour VIsual). Cet éditeur est un héritage de la version BSD d'UNIX.

C'est un programme rescapé des terminaux non graphiques (mode caractère) qui ne nécessite pas l'utilisation d'une souris pour éditer du texte. Les seules touches du clavier permettent de mettre en œuvre les nombreuses fonctionnalités de l'éditeur.

Cependant beaucoup d'utilisateurs rebutent à utiliser cet éditeur même s'il offre des fonctionnalités puissantes pour aider les programmeurs.

L'administrateur, en cas de nécessité, va pouvoir à l'aide de cet éditeur, soit modifier des fichiers de configuration des services du système d'exploitation, soit écrire des scripts d'administration.

L'administrateur peut être aussi obligé de s'en servir dans les plus mauvais moments, c'est-à-dire lorsque plus rien d'autre ne fonctionne (interface graphique par exemple).

Dans le cas où l'utilisateur travaille directement sur la station UNIX/LINUX, il peut disposer d'une interface graphique et donc d'un éditeur graphique associé à l'interface fournie.

Dans le cas fréquent où l'utilisateur (voire l'administrateur) se connecte sur la station UNIX/LINUX depuis sa station bureautique, par exemple en émulant un terminal sous Windows, il n'aura pas accès à l'interface graphique. Alors l'éditeur VI reste le seul moyen d'opérer sur la station.

L'éditeur VI, quant à lui, est présent sur toute version d'UNIX/LINUX.

Toutes ces raisons non exhaustives tendent à prouver l'utilité d'un tel éditeur.

S'il s'avère utile, quelles sont les limites d'un tel éditeur aussi rustique ? L'auteur du document a testé avant d'en avoir la confirmation écrite qu'il est limité aux fichiers de 64Mo ou de 1.100.000 lignes. Il existe d'autres limites mentionnées dans l'aide en ligne.

Ce document est écrit pour aider les utilisateurs débutants dans la mise en œuvre de l'éditeur VI, il se limite à une énumération partielle de ses fonctionnalités les plus communes ou les plus utiles.

Le mot VI se prononce conventionnellement "vi ai".

1.2 Fonctionnement

L'éditeur fait partie de la famille ancienne des éditeurs dits de pleine page. Cela signifie qu'une fois lancé, il occupe l'intégralité de l'écran du terminal. Seule la fermeture de l'éditeur permet à l'utilisateur de retrouver l'invite de commandes.

Si le fichier édité ne remplit pas l'écran, il effectue du remplissage à l'aide du signe « ~ » en début de ligne.

L'éditeur VI utilise principalement deux modes de travail, le mode "commande" et le mode "insertion".

Son fonctionnement est lié à la définition du terminal sur lequel il est exécuté.

Il est possible de changer le comportement de VI, lors d'une session ou de manière durable, en modifiant ses variables d'environnement.

Il utilise pendant une session un fichier temporaire (sous /tmp) et un tampon d'édition.

1.2.1 Lancement

L'éditeur VI se lance soit directement sur un terminal en mode caractère, soit dans un environnement graphique, soit via le réseau (administration à distance) via un accès ssh (secure shell).

Pour lancer l'éditeur, saisir la commande :

vi [-option] NomDuFichier

Comme pour toute commande, il faut faire précéder le nom du fichier par le chemin de recherche.

Remarques :

- Si le fichier existe à l'endroit mentionné par le chemin, celui-ci est lu par VI et le curseur se place au début de la première ligne.
- Si le fichier n'existe pas, une page vide est affichée à l'écran, chaque début de ligne étant signalée par le caractère "~" (tilde).
- Lorsqu'on ouvre un fichier, VI le charge dans un tampon (buffer) et l'affiche à l'écran. Lorsque le fichier est sauvegardé, VI enregistre le tampon sur le disque en écrasant l'ancien fichier si celui-ci existait.

Options :

La syntaxe indique que l'éditeur dispose d'options. Voici quelques-unes des plus usitées :

-r : Liste les fichiers sauvegardés en zone de swap après un arrêt brutal de l'éditeur (coupure de **courant par exemple**),

-R : En lecture seulement,

+numéro : Amène le curseur sur la ligne indiquée par le numéro (sert au débogage),

+/chaîne : Amène le curseur au début de la ligne contenant la première occurrence de la chaîne.

Avertissements : Éviter d'éditer simultanément le même fichier sur deux terminaux. Sous LINUX, le deuxième fichier ouvert est un fichier caché (premier caractère égal à ".") Avec extension ".swp". Il est conseillé de fermer l'éditeur, en le forçant s'il le faut. Puis il faut supprimer la copie du fichier à extension ".swp" pour retrouver un fonctionnement normal.

1.2.2 Modes de travail

Il existe deux modes principaux de travail dans VI :

- mode commande
- mode insertion (ou édition)

La philosophie de VI est d'alterner entre ces deux modes de fonctionnement.

Au lancement de VI, on est en mode commande.

Pour insérer du texte, il faut utiliser une commande pour passer en mode d'édition, puis ajouter le texte voulu. Une fois l'édition terminée, l'utilisateur doit acquérir l'automatisme de revenir en mode commande en appuyant sur la touche Echap, quel que soit le mode en cours. Si nous étions déjà en mode commande, une sonnerie d'avertissement se fait entendre.

Les commandes utilisées dans VI sont des combinaisons de touches du clavier. En mode commande, chaque touche a une signification autre que la valeur de la lettre. Par exemple, « i » devient une commande de passage en mode édition (insertion à partir de la position du curseur).

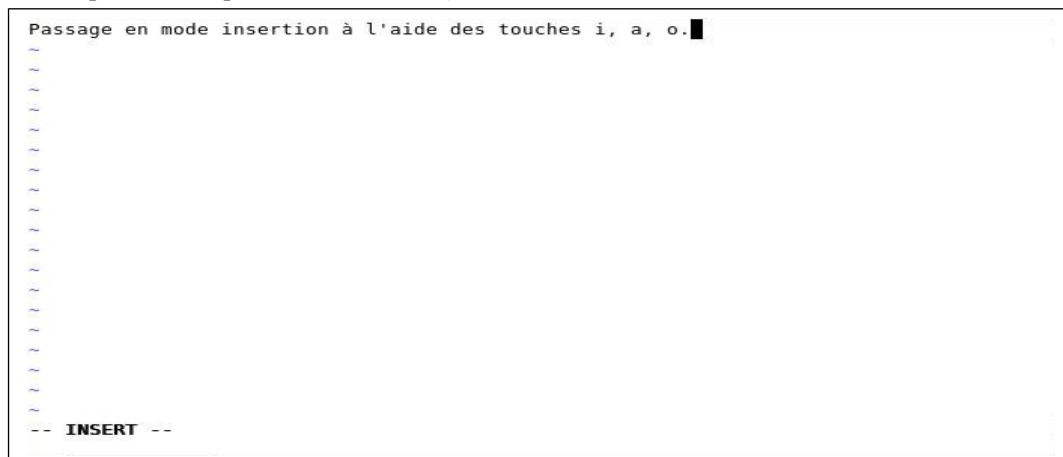


Figure 1

Attention : l'éditeur VI différencie les minuscules des majuscules ! Ceci peut créer des incidents de manipulation, mais surtout augmente le nombre de commandes disponibles.

Remarques :

- Par défaut, pour renseigner l'utilisateur sur le mode courant, l'éditeur affiche celui-ci en bas à gauche de l'écran (figure 2). S'il n'est pas affiché, cela veut dire que l'utilisateur est en mode commande.
- L'affichage du mode courant correspond à une préférence de fonctionnement de l'éditeur. Celles-ci seront vues dans la section "VARIABLES" (paramètre showmode).
- Si le commutateur showmode est activé, l'éditeur affiche en bas à gauche le mode d'insertion : INSERT, APPEND ou REPLACE (remplacement), ou encore VISUAL (inverse vidéo) pour la version VIM.

1.2.3 Cas pratique

Si les explications qui viennent d'être données paraissent obscures, nous allons éclaircir tout cela avec quelques exemples.

Commençons donc par éditer un nouveau fichier : **vi bienvenue**



Figure 2

Nous voici donc dans VI et en mode commande. On voit apparaître en bas de l'écran un message « bienvenue » [New File], indiquant qu'il s'agit d'un nouveau fichier. À présent, on passe en mode insertion avec la commande « i » puis on saisit un texte quelconque. Enfin on repasse en mode commande avec la touche Echap. Il est possible de sauvegarder la saisie effectuée à l'aide d'une commande ex :w (figure 3).

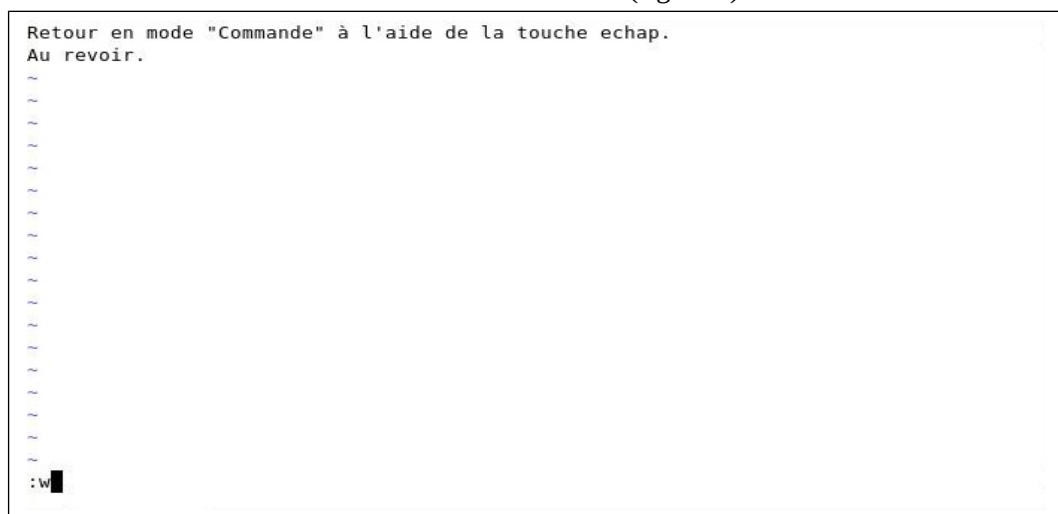


Figure 3

L'enregistrement réalisé, l'éditeur nous renseigne en bas de l'écran : 2 lignes, 67 caractères (figure 4).

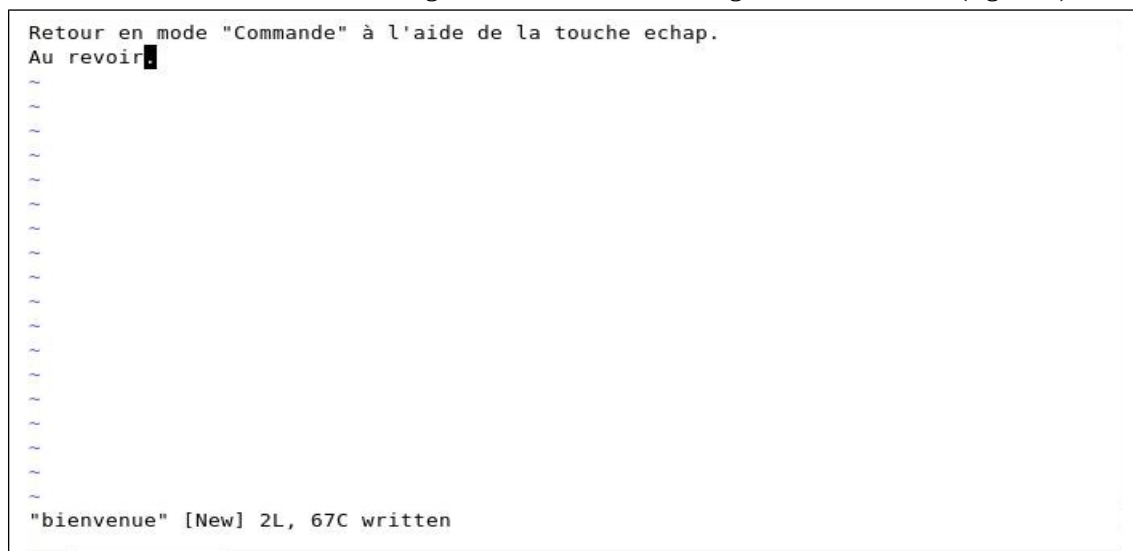


Figure 4

On peut, à présent, continuer à insérer du texte. En mode insertion, on peut ajouter une ligne en validant la touche Entrée, supprimer du texte avec la touche "retour arrière", se déplacer avec le pavé de flèches du clavier.

Une fois l'édition terminée, on enregistre les changements et quitte l'éditeur en une seule commande ZZ. Notons qu'il s'agit de majuscules et que les caractères n'apparaissent effectivement pas. Nous voici revenus à l'invite de commandes du shell.

1.3 Les commandes

Cette section présente une liste conséquente des commandes à utiliser uniquement en mode commande.

1.3.1 Conventions d'écriture

[option] désigne les parties facultatives d'une commande.

[n] signifie qu'un nombre peut précéder la commande.

{expression variable} désigne la valeur de l'argument qui suit la commande.

Voici quelques abréviations de caractères les plus couramment utilisées :

<Echap> caractère d'échappement

<Espace> caractère "espace" ou barre d'espacement

↵ touche "Entrée"

^ touche "Ctrl"

1.3.2 Les commandes

Ces commandes sont regroupées par groupes de fonctionnalités au sein de tableaux : déplacement du curseur, recherche, passage en mode d'édition, effacement, remplacement, transfert de texte, tampons d'édition, commandes ex, fin de session, etc.

1.3.2.1 Déplacement du curseur

Le déplacement à l'aide des touches hjkl fonctionne sur tous les terminaux, mais le déplacement à l'aide des flèches lui est préféré.

Commande	Signification
← ou h	Déplacement du curseur d'un caractère vers la gauche.
↓ ou j	Déplacement d'une ligne vers le bas.
↑ ou k	Déplacement d'une ligne vers le haut.
→ ou l	Déplacement du curseur d'un caractère vers la droite.
-	Déplacement du curseur au début de la ligne précédente.
+	Déplacement du curseur au début de la ligne suivante.
\$	Déplacement du curseur en fin de la ligne.
0	Déplacement du curseur en début de ligne.
b	Déplacement du curseur d'un mot vers la gauche.
w	Déplacement du curseur d'un mot vers la droite (en début de mot).
e	Déplacement du curseur sur le dernier caractère du mot courant.
B, W, E	Mêmes fonctions que b, w, e en ignorant la ponctuation.
M	Déplacement du curseur jusqu'au milieu de l'écran.
L	Déplacement du curseur jusqu'à la dernière ligne de texte.
H	Déplacement du curseur jusqu'à la première ligne de texte.
Page down ou ^F	Page suivante (Forward a page).
Page up ou ^B	Page précédente (Backup a page).
%	Si le curseur était placé sur une parenthèse ou une accolade, qu'elle soit ouvrante ou fermante, cette commande place le curseur sur le signe correspondant (commande utilisée par les programmeurs pour déboguer).
n>>	Indentation de n lignes à partir de la ligne courante.
n<<	Désindentation de n lignes.
:0 ↵	Déplacement vers la 1 ^{ère} ligne du fichier, ou la dernière ligne (: \$ ↵).
: [n] ↵	Déplacement vers la <n> ^{ème} ligne (utile pour les programmeurs en phase de débogage).

1.3.2.2 Insertion

Commande	Signification
a	append : ajouter à la suite. Le texte est inséré immédiatement après la position du curseur.
A	Le texte est inséré à la fin de la ligne courante.
i	insert : insérer. Le texte est inséré immédiatement avant la position du curseur.
I	Le texte est inséré au début de la ligne courante.
o	open new line : ajouter nouvelle ligne. Une nouvelle ligne est insérée après la ligne courante.
O	Une nouvelle ligne est insérée avant la ligne courante.
Echap	Sort du mode insertion.

1.3.2.3 Recherche

D'une manière générale, les chaînes de caractère indiquées en argument derrière des commandes sont désignées sous le terme général d'expressions régulières.

Ces expressions régulières sont des modèles de chaîne qui, plutôt que de définir complètement un texte, n'en donnent qu'une forme générique pour signifier des expressions.

Ils contiennent donc des caractères génériques de substitution dûment interprétés par le système : ce sont les caractères spéciaux ou méta caractères tels les caractères "*" ou ".".

En dehors de ces caractères, le système met à note disposition d'autres caractères comme les crochets pour fixer avec précision les critères de recherche.

Pour définir le critère de recherche, l'utilisateur utilise un ou plusieurs modèles ou motifs (pattern en anglais).

Commande	Signification
/ {chaîne} ↵	Recherche vers la fin du fichier de l'occurrence "chaîne".
? {chaîne} ↵	Recherche vers le début du fichier de l'occurrence "chaîne".
n	Répétition de la dernière recherche.
N	Répétition de la dernière recherche, mais dans l'autre sens.
/^ {str} ↵	Cherche la prochaine ligne qui commence par la chaîne <i>str</i> .

Voici quelques exemples concernant la recherche de :

- l'article "de" ou "De" : **/[dD]e**
- tous les mots de 2 lettres commençant par une majuscule et finissant par "e" en définissant une plage de caractères "A-Z" : **/[A-Z]e**
- tous les caractères exceptés les minuscules : **/[^a-z]**
- tous les mots de 4 lettres finissant par "oto", le point représentant un caractère quelconque : **/.oto**
- tous les mots finissant par "to" précédés de N fois la lettre "o" : **/o*to**
- tous les mots dont la ligne commence (ou finit) par "to" : **/^to** (ou **/to\$**)

Dans certaines situations, ce sont ces caractères spéciaux qu'il faut rechercher dans le texte. Il faut au préalable verrouiller (ou désactiver) leur fonction spéciale. Pour cela, il suffit de placer le caractère spécial "\" devant celui qui est à rechercher.

Voici un exemple où l'on cherche une étoile derrière une lettre "a" : **/a***

1.3.2.4 Remplacement de texte

Commande	Signification
R	Remplace le reste de la ligne courante à partir du curseur.
cw {texte} <Echap>	Change word : changer un mot. Remplace le mot en commençant à la position courante du curseur. Il est nécessaire

	d'appuyer sur <Echap> pour revenir en mode commande.
c0 ou c\$	Remplace depuis la position du curseur vers le début (ou fin) de la ligne.
cf{car}	Remplace depuis la position du curseur jusqu'au 1 ^{er} caractère "car" trouvé sur la ligne.
c/{mot}	Remplace jusqu'à la 1 ^{ère} chaîne "mot" trouvée (signe "\$" apparent).
cc	Remplace la ligne entière.
:s/{chaîne1}/{chaîne2}/ [option] ↵	substitute : substituer. Remplace la 1 ^{ère} occurrence « chaîne1 » par « chaîne2 » dans la ligne courante. L'option g (=global) remplace toutes les occurrences "chaîne1" d'une ligne. L'option c (= confirm) demande confirmation avant remplacement. Remplace la 1 ^{ère} occurrence trouvée de chaque ligne comprise entre les lignes numéro x et numéro y. Le "." indique la ligne courante.
:x,y/...	

1.3.2.5 Effacement de texte

Les touches ou <Suppr> ne fonctionnent généralement pas sous l'éditeur VI basic des UNIX. Elles fonctionnent sous LINUX car « VI » a été remplacé par « Vim » (VI Improve). En revanche, la touche "Retour arrière" ou "Backspace" (bs) fonctionne. Dans ce cas, le mot REPLACE peut apparaître en bas de l'écran le temps du remplacement.

Commande	Signification
[n]x	Effacement de n caractères à partir de la position le curseur.
[n]X	Effacement de n caractères précédant la position du curseur.
[n]dd	Effacement de n lignes (correspond à l'opération couper, le contenu n'est pas perdu).
[n]dw (ou [n]dW)	Supprime n mots (en ignorant la ponctuation).
d/{str} ↵	Efface tout depuis le curseur jusqu'à la première occurrence de str.
db	Effacement jusqu'au mot précédent.
d^ ou d0	Efface de la position actuelle du curseur jusqu'au début de la ligne.
d\$ ou dG	Efface de la position actuelle du curseur jusqu'au à la fin de la ligne.
D	Efface la ligne à partir de la position courante du curseur.
[n]s	Supprime n caractères et passe en mode insertion.

1.3.2.6 Transfert de Texte

L'éditeur VI dispose pour toutes les commandes de suppression ou de transfert de texte d'un tampon principal ou registre temporaire.

Commande	Signification
[n]yy	yank : capturer Copie la ligne courante dans un tampon sans l'effacer
y\$	Met la fin de la ligne dans le tampon.
p	put buffer : récupérer tampon Insertion du contenu du tampon après le curseur (opération "coller") S'il s'agit d'une ligne capturée par yy ou dd, celle-ci est insérée à partir de la ligne suivant celle du curseur. Si le tampon contient un mot, celui-ci est inséré immédiatement après le curseur.
P	Put buffer : récupérer tampon Idem que la commande précédente, mais le contenu du tampon est inséré avant.
:15,17m23 ↵	Déplacer les lignes 15 à 17 en ligne 23 (cette dernière doit exister).
:10,18co25 ↵	Copie les lignes 10 à 18 en ligne 25.

1.3.2.7 Registres nommés

L'éditeur VI, outre le tampon principal, dispose de 36 tampons ou registres nommés pour conserver des morceaux de texte (26 pour les lettres, 10 pour les chiffres).

Peu d'utilisateurs utilisent les autres tampons. Le bloc de texte peut être placé dans un autre tampon demandé en utilisant la commande « " ». Après avoir tapé « " », une lettre ou un chiffre identifiant le tampon doit être entré. Voici quelques exemples.

Commande	Signification
"ayy	Met la ligne courante dans le registre "a".
"bdd	Détruit la ligne courante et la range dans le registre "b".
"ap	Insère le contenu du registre "a" après la ligne courante.
"bP	Insère le contenu du registre "b" avant la ligne courante.

Remarque : si l'utilisation des tampons « lettre » est aisée, celui des tampons « chiffre » l'est moins.

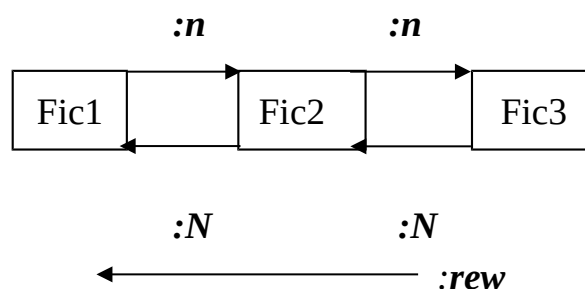
1.3.2.8 Commandes "ex" (ou de bas de page)

On passe du mode commande au mode bas de page en tapant sur ":" ; ce qui a pour effet de placer le curseur au bas de l'écran. Toutes les commandes ex doivent être validées par ↵

Commande	Signification
:f	Affiche le nom du fichier actif et le nombre de lignes.
:w%[n]	Le caractère % possède une signification particulière. En effet, % représente le nom courant du tampon, c'est à dire le nom du fichier ouvert au lancement de l'éditeur. Ainsi, en utilisant la commande :w%2, les modifications sont enregistrées sous le nom fichier2 dans le répertoire courant.
:w[!]<espace>[fic]	Le fichier courant est écrit sur le disque. Un nom de fichier en option permet d'enregistrer sous...(ou :file nom).
:x,yw<espace>{fic}	Écriture des lignes x à y dans le fichier « fic ».
:x,yw>>{fichier}	Ajoute les lignes x à y à la fin du fichier « fichier ».
:r{fichier}	Insère le contenu du fichier indiqué en argument à partir de la ligne qui suit le curseur (ou commande :read).
:r<espace>!cmd	Insère le résultat de la commande (ou !! cmd).

L'option "!" employée derrière la commande force l'éditeur à exécuter la commande, malgré un avertissement.

Il est possible d'éditer plusieurs fichiers. Il existe des commandes ex de gestion des différents fichiers ouverts. Les commandes "copier" (yy), "couper" (dd) et "coller" (p) sont disponibles d'un fichier à l'autre.



:n<espace>{fichier}	Ouvre le 2 ^{ème} fichier fichier. Sans arguments, permet de basculer d'un fichier édité au suivant (voir schéma).
:N	Permet de basculer d'un fichier édité au précédent (voir schéma).
:arg	Montre la liste des fichiers édités, et met entre crochets celui qui est actif à l'écran.
:rew	Retourne au 1 ^{er} fichier édité.

Sans quitter l'éditeur, il est possible aussi de lancer une commande, d'écrire un script shell (:sh), d'exécuter une commande sur une expression régulière donnée (:x,y/str/cmd), de trier le fichier en cours d'édition (:%!sort), etc.

:{cmd}	Exécute une commande sans sortir de VI
--------	--

:set<espace>[option]	Préférences de fonctionnement de l'éditeur. Voir section VARIABLES.
----------------------	---

1.3.2.9 Fin de session VI

L'ajout du "!" derrière les commandes suivantes permet de forcer l'éditeur en cas d'incident, ou d'édition d'un fichier non accessible en écriture.

Commande	Signification
:x[!]	Sortie avec sauvegarde.
:w[!]	Sauvegarde sans sortie.
:q[!]	Sortie sans sauvegarde.
ZZ	Sortie avec sauvegarde.

1.3.3 Commandes diverses

Commande	Signification
~<espace>	Change les minuscules en majuscules (ou vice versa).
u	undo : défaire Annule la dernière commande qui a modifié le contenu du tampon.
U	Annulation de toutes les commandes de modification de texte affectant la ligne.
J	Join lines : Joindre lignes. Jonction de la ligne courante et de la ligne suivante. J supprime le caractère de fin de ligne <nl>, les deux lignes sont jointes.
^G	Affiche le nom du fichier, le N° de ligne courante, le nombre de lignes totales, le pourcentage d'affichage.

1.4 Les variables

L'éditeur VI possède un certain nombre de variables d'environnement paramétrables ou options que l'on peut positionner pour produire des effets spéciaux.

Ces options se présentent sous deux formes :

- les variables à 2 états : actifs ou non (ou bascules internes, ou commutateurs, ou switches),
- les variables numériques ou alphanumériques.

Les options à deux états sont :

- activées par :set [option] ↵
- désactivées par :set no[option] ↵

Le premier exemple que l'on peut fournir concerne la numérotation des lignes avec l'option nu :

- Pour installer la numérotation :set nu
- Pour la retirer :set nonu

Les options à valeurs numériques ou alphanumériques sont activées par une commande du style :

:set option=valeur ↵

Le premier exemple fourni concerne la valeur du taquet de tabulation (ici 4 colonnes) :

:set tabstop=4.

Pour afficher la table de tous les paramètres, il faut saisir :

:set all

Ces valeurs sont écrites dans un fichier de configuration associé à l'éditeur. Sous LINUX, ce fichier se nomme **.vimrc** (éditeur Vim). Le fichier **.vimrc** définit aussi les associations de touches.

Les commandes passées au cours d'une session (:set) ne restent valides que le temps de la session. Il est possible de modifier les valeurs par défaut pour le prochain lancement de VI.

Pour cela, plusieurs méthodes sont proposées :

- Utiliser une variable d'environnement du système d'exploitation, modifier le fichier de configuration actuel (.vimrc sous LINUX)
- Ou lui substituer un autre fichier de configuration de remplacement (.exrc).

La première méthode consiste à placer la commande set appropriée dans la variable EXINIT et l'exporter dans votre environnement (cas des shells sh ou ksh).

Exemple:

```
EXINIT='set ai aw terse sh=/bin/ksh'
export EXINIT
```

Ces instructions sont généralement placées dans le fichier **.profile** ou **.login** propre à l'utilisateur connecté.

Dans le cas d'un shell **csh**, la commande "setenv EXINIT" place les paramètres dans le fichier **.cshrc**.

Une autre possibilité de modifier ces valeurs consiste à placer ces instructions dans un fichier **.exrc** (commandes ex) qui doit se trouver dans le répertoire de connexion de l'utilisateur.

Exemple de configuration d'un fichier .exrc

```
set nu ai ts=3 showmode      # numérotation des lignes, indentation et tabulation et affichage du mode
                              d'insertion
:ab esiea "École Supérieure Informatique Électronique Automatique"
# remplacement du sigle « esiea » par "École Supérieure..."
```

1.4.1 Commutateurs

La commande « set » peut s'utiliser aussi bien avec la valeur littérale que la valeur abrégée.

Valeur littérale	Abréviation	Signification
autoindent	ai	Pour aligner sur la ligne précédente, ^D pour annuler.
autowrite	aw	Mise à jour avant changement de fichier.
beautify	bf	Ignore certains caractères de contrôle.
directory=/tmp	dir	Répertoire des fichiers temporaires.
edcompatible		Compatibilité avec l'ancien éditeur 'ed'.
errorbells	eb	Signal sonore sur erreur.
flash		Flash d'écran en cas d'erreur
hardtabs=8	ht	Longueur de taquet matériel
ignorecase	ic	Confond majuscules/minuscules dans une recherche.
list		Affiche les caractères invisibles (tabulation ^I, fin de ligne \$).
magic		Neutralisation automatique.
mesg		Affichage des messages de l'utilitaire write.
number	nu	Numérotation des lignes.
noreadonly		Lecture seulement.
redraw		Simule un terminal intelligent.
ruler	ru	Affiche les coordonnées du curseur (pas sur HP-UX).
scroll=11		Taille de la demi-fenêtre.
shell=/bin/sh	sh	Interpréteur par défaut interne à l'éditeur.
shiftwidth=8	sw	Largeur des retraits de ligne.
noshowmatch	sm	Montre la parenthèse ou l'accolade correspondante à l'une d'entre elles (fermeture ou ouverture).
noshowmode	smd	Affichage du mode de travail (INSERT ou COMMAND), ou du mode d'insertion (INSERT, APPEND, REPLACE). Ne fonctionne pas avec tous les types de terminaux.
tabstop=8	ts	Valeur du taquet de tabulation en colonnes.
taglength=0	tl	Longueur des marquants.

Valeur littérale	Abréviation	Signification
tags=tags /usr/lib/tags		Localisation des marquants.
term=vt220		Type de terminal utilisé.
noterse		Messages d'erreur courts.
ttytype=vt220		Type de terminal utilisé.
visualbell	vb	Inverse le BEEP par un effet visual.
warn		Avertissement si sortie sans sauvegarde, ou sortie forcée (!).
window=23		Taille de la fenêtre en nombre de lignes.
wrapscan		Autorise les recherches circulaires.
wrapmargin=0	wm	Distance du bord droit de la fenêtre à partir duquel le retour à la ligne se fait.

1.5 Autres éditeurs

Malgré son apparence, l'éditeur **VI** n'était pas le premier conçu pour le monde UNIX. Il reprenait les conceptions d'autres éditeurs tels que « **ed** » (commandes de recherche d'expressions régulières) et « **ex** » (commandes de bas de page).

Les éditeurs "ed" et "ex" sont des éditeurs ligne par ligne qui ne réagissent qu'aux commandes entrées du clavier sans présenter le fichier de façon visuelle comme le fait VI.

Il existe un autre éditeur "sed" (Stream editor) qui lit un fichier d'entrée (stdin) et produit un fichier de sortie (stdout). Les éditeurs "ed", "ex" et "sed" sont généralement utilisés dans des scripts, alors que l'éditeur VI sert plutôt de manière interactive.

Les éditeurs "ed" et "VI" sont intimement liés. Il est possible d'éditer un fichier sous "ed" puis de basculer sous VI en cours de session ed (:vi).

Chaque version d'UNIX peut présenter son propre éditeur.

1.5.1 VIM

L'éditeur VI d'origine n'est plus présent sur les distributions LINUX. La version VIM (Vi IMproved) l'a depuis longtemps remplacé (www.vim.org). Elle a apporté entre autres l'annulation multiniveaux (undo), la multiédition et les 36 tampons, l'utilisation des touches de flèches du clavier, l'alignement du texte et la coloration syntaxique.

Sans quitter l'éditeur, la commande **:help** permet de consulter l'aide en ligne.

Ces distributions proposent bien sûr des versions graphiques telles que Vi étendu vim dont voici une copie d'écran de la barre de menus et d'icônes.

Les différentes icônes correspondent aux fonctionnalités les plus usitées.

1.6 En conclusion

Le document présent reste axé principalement sur l'éditeur VI. Aussi se doit-on d'expliquer, dans un premier temps, au jeune informaticien, les raisons cachées qui poussent les utilisateurs d'UNIX avertis (administrateurs et programmeurs) à l'apprécier.

Enfin et surtout, le document ouvre la porte aux éditeurs graphiques. Ils ont toujours existé, mais le manque de standard dans ce domaine fait qu'il existe autant d'éditeurs que d'interfaces graphiques.

Malgré tout, il est conseillé au jeune utilisateur d'UNIX de se faire une raison, et de s'acclimater peu à peu à l'usage de l'éditeur VI en pratiquant des exercices fréquents. Il reste le seul éditeur que l'on est sûr de trouver sur toutes les versions d'UNIX.