

## AIDE-MEMOIRE VI

**Vi** est un éditeur de textes plein écran présent sur la grande majorité des systèmes UNIX™. Voici un aide-mémoire pour son jeu de commandes de base.

Des commandes supplémentaires peuvent toutefois exister sur votre propre éditeur **vi** ou compatible. Reportez-vous à la documentation de votre système.

### Notations

**x** ou **X** : caractère tapé (minuscule ou majuscule)

**C-x** : caractère tapé en maintenant Ctrl appuyée.

Les caractères spéciaux seront notés par leur nom abrégé : **«Ech»** pour échappement, **«Ent»** pour entrée, **«Ret»** pour retour arrière...

Les commandes commençant par **:**, **/**, ou **?** s'affichent automatiquement au bas du terminal (ligne d'état) et doivent être terminées par **«Ent»**.

**[ ]** : facultatif; ne pas taper les crochets

**car** : à remplacer par un seul caractère

**chn** : à remplacer par une chaîne de caractères

**fich** : à remplacer par un nom de fichier

**cmd** : à remplacer par une commande

**n** : à remplacer par un nombre (suite de chiffres)

**nn** : à remplacer par une référence à un numéro de ligne :

⇒ un nombre (1, 25, ...)

⇒ un marqueur ou un déplacement (., \$, ...)

⇒ une expression à rechercher ( / chn / )

### Appel de l'éditeur

**vi [fich]** : Edite le fichier *fich*, s'il existe. Sinon, édite un nouveau fichier vide.

**vi -r fich**: Edite la dernière version du fichier *fich* en cas d'arrêt inopiné du système ou de l'éditeur.

**vi +[nn] fich** : Edite le fichier *fich* avec le curseur à la ligne *nn* [ par défaut : à la dernière ligne ]

**vi fich1 fich2...** : Edite chacun des fichiers. Après avoir sauvé les changements dans l'un des fichiers, taper **:n «Ent»** pour passer au suivant.

### Modes d'édition

**Mode insertion** : Les caractères tapés s'insèrent dans le texte du fichier édité. Un jeu réduit de commandes est disponible par l'intermédiaire de caractères de contrôle. Ce mode se termine en tapant **«Ech»**.

**Mode commande** : Les caractères tapés sont considérés comme des commandes d'édition de texte.

Les commandes en majuscules et minuscules diffèrent. Il faut vérifier que les majuscules ne sont pas verrouillées si l'éditeur présente un comportement inhabituel.

## Manipuler des fichiers

**:q** : quitte vi  
**:q!** : quitte vi sans sauvegarder les changements  
**ZZ**, **:wq!** ou **:x** : sauve si possible et quitte vi

**:[nn,nn]w[!]«Esp»[>>]fich** : sauve le texte dans *fich*  
**:w** : sauve sous le nom de fichier courant  
**:w!** : sauve sans vérifier les droits d'écriture  
**:w >>fich** : ajoute le texte à la fin de *fich*  
**:nn,nnw fich** : sauve les lignes de l'intervalle *nn, nn*

**:args** : affiche la liste des fichiers à éditer  
**:n[!]** : édite le fichier suivant  
**:n!** : ignore les changements  
**:rew[!]** : édite le premier fichier dans la liste  
**:f fich** : change le nom de fichier courant

**:e[!][+n] [fich]** : édite le fichier *fich* - voir **:w**  
**:e!+3 fich** : édite *fich* en ligne 3, ignore changements  
**:e#** : édite le dernier fichier

**:[nn]r fich** : insère *fich* dans le texte édité  
**:[nn]r !sh-cmd** : exécute une cmd shell et insère sa sortie dans le texte (Si *nn*, après la ligne *nn*, sinon après le curseur)

**:cd [répertoire]** : change de répertoire de travail

## Déplacer le curseur

Chacune de ces commandes peut être précédée d'un nombre d'exécutions : **12w** avancera de 12 mots.

**h** ou ← ou **C-h** ou «Ret» : un caractère à gauche  
**l** ou → ou «Esp» : un caractère à droite  
**j** ou ↓ ou **C-n** : un caractère vers le bas  
**k** ou ↑ ou **C-p** : un caractère vers le haut  
**«Ent»** ou +, - : début de ligne suivante, précédente  
**w, b, e** : mot suivant, mot précédent, fin du mot  
**W, B, E** : pareil sans tenir compte de la ponctuation  
) , ( : phrase suivante, précédente  
} , { : paragraphe suivant, précédent  
]] , [[ : section suivante, précédente (v. [Marquer le texte](#))

**\_** ou **^** : avance jusqu'au prochain non-blanc  
**fcar** : avance jusqu'au prochain caractère *car*  
**Fcar** : revient au dernier caractère *car*  
**tcar** : avance jusqu'au caractère précédant *car*  
**Tcar** : revient au caractère suivant *car*  
**'car** : premier non-blanc de la ligne marquée *car*  
**`car** : va au caractère marqué *car* (v. [Marquer le texte](#))

Les commandes suivantes peuvent aussi être précédées par *n*, mais le résultat dépend de la commande.

**G, nG** ou **:n** : fin du texte, ligne *n* du texte  
**H, nH** : début de l'écran, ligne *n* de l'écran  
**L, nL** : fin de l'écran, *n* lignes avant la fin de l'écran  
**O, \$** : début de la ligne, fin de la ligne  
**n\$** : fin de la (*n-1*)ème ligne après la ligne courante  
**|, n|** : première colonne, *n*-ième colonne  
**/str[/]«Ret»** : Recherche vers l'avant de l'expression *str*  
**?str [?]«Ret»** : Recherche vers l'arrière de l'expression *str*

## Ajuster l'écran

**C-l** : effacer et redessiner l'écran

**C-e, C-y** : déplacer l'écran d'une ligne vers le haut, vers le bas  
**C-u, C-d** : déplacer l'écran d'une demi-page vers le haut, le bas  
**C-b, C-f** : déplacer l'écran d'une page vers le haut, vers le bas  
Ces commandes précédées par *n* s'exécutent *n* fois  
ex : **3C-b** remonte de trois pages dans le texte

**z «Ent»** : mettre la ligne courante en haut de l'écran  
*nnz «Ent»* : mettre la ligne *nn* en haut de l'écran  
*nnz.* : mettre la ligne courante [ ou *nn* ] au milieu de l'écran  
*nnz-* : mettre la ligne courante [ ou *nn* ] en bas de l'écran

## Modifier le texte

### - Passage en mode insertion:

Si *n* est tapé avant la commande, le texte inséré est répété *n* fois.

**a** : insère du texte après le curseur, fin par «Esc»  
**A** : insère du texte après la fin de la ligne  
**i** : insère du texte avant le curseur  
**I** : insère du texte avant le premier non-blanc de la ligne  
**o, O** : ouvre une ligne avant, après la ligne du curseur

### - Commandes accessibles en mode insertion:

«Esc» : retour en mode commande  
«Ent» : passe à la ligne  
«Ret» ou **C-h** : efface le dernier caractère  
**C-v car** : permet d'insérer un car. de cmd («Esc», **C-h**, **C-@**,...)  
**C-@** : répète la dernière insertion si 1<sup>er</sup> caractère du mode  
**C-w** : efface le mot précédent  
**C-t** : indente d'une position à droite (v. option **sw**)

### - Mode commande:

Opérateurs acceptant une commande de déplacement :  
[*n*]oper[*n*][*dépl.*] (v. **Déplacer le curseur**) - oper peut être :  
**d** : coupe jusqu'au curseur déplacé **dd** : la ligne  
**y** : copie jusqu'au curseur déplacé **Y** ou **yy** : la ligne  
**c** : change jusqu'au curseur déplacé **cc** : la ligne  
ex.: **d3w** coupe les 3 mots suivants, **4d3w** coupe les 12 mots suivants  
(4 fois 3 mots), **4yy** copie les 4 lignes suivantes

### Autres opérateurs :

**p, P** : colle après, avant le curseur  
[*n*]**x**, [*n*]**X** : coupe *n* caractères après, avant le curseur  
**D, C** : coupe ou change jusqu'en fin de ligne (v. **d\$** ou **c\$**)  
~ : change le caractère sous le curseur maj. <-> min.  
[*n*]. : répète le dernier changement [ *n* fois ]  
[*n*]r**car** : remplace le caractère sous le curseur par *car* [ *n* fois ]  
[*n*]R**str «Esc»** : remplace le texte à partir du curseur par *str* [ *n* f. ]  
[*n*]S**str «Esc»** : substitue *str* au caractère sous le curseur [ *n* car. ]  
[*n*]S**str «Esc»** : substitue *str* à la ligne [ *n* lignes à partir du curseur ]  
[*n*]J : concatène deux lignes [ *n* lignes ]  
[*n*]<<, >> : indente de *n* car. à gauche, à droite (v.opt. **sw**)  
:[*nn*,*nn*]**m n** : déplace l'intervalle ou la ligne après la *n*-ième ligne  
:[*nn*,*nn*]**t n** : copie l'intervalle ou la ligne après la *n*-ième ligne

## Annuler et Répéter

**u** : Annule la dernière modification  
**U** : Annule les dernières modifications de la ligne courante  
**.** : répète le dernier changement

L'utilisateur dispose de

- 26 tampons de texte nommés de **a** à **z**
- 9 tampons des suppressions successives nommés de **1** à **9**

**p, P, y, Y** et **d** peuvent être précédées par `"car (minuscule)` pour utiliser ou remplacer le tampon nommé `car`  
**y, Y** et **d** peuvent être précédées par `"car (majuscule)` pour rajouter le texte copié/collé dans le tampon `car`

`"car p` : rappelle le contenu du tampon nommé `car`  
`"1pu.u.u.` : recherche parmi les dernières suppressions

## Marquer le texte

`mcar` : marque la position du curseur avec `car`  
``car` : retourne au caractère marqué `car`  
```` : retourne à la ligne d'origine du dernier déplacement  
`'car` : retourne au 1er non-blanc de la ligne marquée `car`  
`"` : retourne au 1er non-blanc de dernière ligne changée

Une phrase se termine par `.`, `!` ou `?` suivi de **«Ent»** ou deux **«Esp»**.

Un paragraphe est délimitée par une ligne vide.

Une section est délimitée par `{` en première colonne.

Les commandes `(, )`, `{, }`, `[, ]` permettent de déplacer le curseur par phrases, paragraphes ou sections. (v. [Déplacer le curseur](#))

## Rechercher et remplacer

**Recherche de délimiteur associé:**

`%` : avec curseur sur `(, )`, `[, ]`, `{, }` ou `}`, va au second délimiteur

**Recherche d'une lettre:** (voir `f, F, t, T`, [Déplacer le curseur](#))

`;` : répète la dernière recherche par `f, F, t, T`

`,` : comme `;` mais en sens inverse

**Recherche et remplacement de chaînes :**

On peut utiliser des *expressions régulières* qui acceptent les caractères spéciaux suivants :

`^` : début de ligne (si 1er caractère de l'expression)

`$` : fin de ligne (si dernier caractère de l'expression)

et si l'option **magic** a été choisie,

`.` : joker, représente un caractère quelconque

`[ str ]` : représente un caractère contenu dans `str`

`[^ str ]` : représente un caractère non contenu dans `str`

`[ c1-c2 ]` : représente un caractère entre `c1` et `c2` inclus

`\<, \>` : représente le début, la fin d'un mot

`*` : cherche 0, 1 ou plusieurs occurrences du dernier car.

`\(, \)` : délimitent une sous-expression pour les remplacements

`&` : remplacé par la chaîne trouvée

`\n` : remplacé par la *n*-ième sous-expression `\( ... \)`

`\car` : remplacé par `car` ( permet d'obtenir `/$^.*[&| - et \)`

#### Recherche de chaîne :

`/str, ?str` : recherche l'expression `str` en avant, en arrière  
`/ «Ent», ? «Ent»` : répète la dernière recherche avant, arrière  
`//` : dernière expression régulière entrée  
`~` : dernière expression de remplacement entrée  
`/str / ; // ; //` : recherche la troisième occurrence de `str`  
`/str /±n` : va à la n-ième ligne après (+) ou avant (-) `str`  
`n` : répète la dernière recherche dans le même sens  
`N` : répète la dernière recherche en sens inverse

#### Remplacement de chaîne :

`:s/str1 /str2 /str3` : substitue `str2` à `str1` dans la ligne  
`str3` est formé à partir de  
`c` : confirmation des substitutions  
`g` : substitue toutes les occurrences de `str1` dans la ligne  
`p` : affiche les changements  
`:[nn,nn]g/str` : va à la dernière ligne contenant `str` [dans `nn,nn`]  
`:[nn,nn]g/str /cmd` : exécute `cmd` sur les lignes contenant `str`  
`:[...]g!/str /cmd` : exécute `cmd` sur les lignes ne contenant pas `str`  
application typique de `:g` :  
`:g/str1/s//str2 /g` : change `str1` en `str2` partout dans le texte  
Le séparateur après `:g` ou `:s` n'est pas obligatoirement `/`  
`:s&/&\\&g` : change tous les `/` de la ligne en `\`

## Macros et abréviations

#### Macros :

`:map str cmd` : exécute `cmd` lorsque `str` est pressé  
`:unmap str` : supprime la définition de la macro `str`  
`:map` : affiche toutes les macros définies

Ces directives s'appliquent en mode commande seulement. Pour créer, supprimer ou afficher des macros valides aussi en mode insertion, il faut utiliser `:map!` ou `:unmap!`.

#### Abréviations:

`:ab str texte` : dès que `str` est inséré, remplace par `texte`  
`:una str` : supprime la définition de l'abréviation `str`  
`:ab` : affiche toutes les abréviations définies

La liste des macros et abréviations d'un utilisateur est souvent regroupée dans le fichier `.exrc` (voir [Configuration](#)). Cela permet par exemple de gérer les touches de fonction, les flèches et autres touches spéciales sur des terminaux non standard.

Les touches inutilisées dans vi sont les touches de fonction et `K V g q v * =`

## Etat du texte

Le résultat s'affiche sur la ligne d'état, en bas de l'écran

`:f` ou `C-g` : affiche le nom de fichier courant et la position  
`:.=` : affiche le numéro de ligne courant (ligne du curseur)  
`:=` : affiche le nombre de lignes du fichier  
`:l` : affiche les caractères spéciaux de la ligne courante

## Commandes 'shell'

### Lancer un interpréteur de commandes:

**:stop** ou **C-z** : suspend vi, retour par **fg** sur BSD  
**:sh** : lance un shell depuis vi, retour par **C-d**

### Exécution d'un script shell:

**:so** *fich* : exécute le script shell *fich*

### Exécution d'une commande:

**:! cmd** : exécute la commande *cmd*, qui peut utiliser  
**%** : remplacé par le nom du fichier courant  
**#** : remplacé par le nom du dernier fichier édité  
**!** : remplacé par la dernière commande lancée par **!**

**:[nn]r! cmd** : insère la sortie de *cmd*  
**:[nn,nn]w! cmd** : exécute *cmd* avec le texte édité en entrée  
Voir aussi **:w** et **:r** dans [Manipuler des fichiers](#) pour utiliser *nn*.

### Filtrage du texte:

**[n]! dépl cmd** : envoie la région de texte définie par *n* et *dépl* (déplacement depuis le curseur courant, voir [Déplacer le curseur](#)) sur l'entrée de la commande *cmd*, puis remplace cette région par la sortie de *cmd*. Exemple :

**4!+sort -n** trie numériquement cinq lignes à partir du curseur  
**!Gexpand** réécrit les tabulations avec des espaces depuis la position du curseur jusqu'à la fin du texte.

## Configurer l'éditeur

### Modifier les options de vi :

**:set opt** : modifie l'option *opt*. Pour une option de nom *xxx*,  
**:set xxx** positionne l'option *xxx* à vrai  
**:set noxxx** positionne l'option *xxx* à faux  
**:set xxx=val** donne une valeur à l'option *xxx*  
**:set xxx?** Affiche la valeur de l'option *xxx*

**:set all** Affiche toutes les options  
**:set** Affiche les options au démarrage et les changements

Voici une liste non exhaustive des options de vi. Une implémentation particulière peut toujours comporter des variantes sur les valeurs par défaut, ou des options non listées ici.

| Nom        | abrév. | défaut   | description                                        |
|------------|--------|----------|----------------------------------------------------|
| autoindent | ai     | noai     | indentation automatique                            |
| autoprint  | ap     | ap       | affiche la ligne après <b>c, d, j, m, :s, t, u</b> |
| autowrite  | aw     | noaw     | sauve automatiquement le texte                     |
| directory  | dir    | dir=/tmp | emplacement du tampon disque                       |
| errorbells | eb     | noeb     | sonne lors des messages d'erreur                   |
| hardtabs   | ht     | ht=8     | position des tabulateurs du terminal               |
| ignorecase | ic     | noic     | maj/min indifférenciées p. recherche               |
| lisp       | -      | nolisp   | modifie <b>()[]{}</b> pour codage en lisp          |
| list       | -      | nolist   | affiche tabulations et fins de lignes              |
| magic      | -      | magic    | étend car. spéciaux en recherche                   |
| mesg       | -      | mesg     | laisse passer les messages <b>write</b>            |
| number     | nu     | nonu     | numérote les lignes                                |
| open       | -      | open     | autorise <b>open</b> avec vi                       |
| optimize   | opt    | opt      | supprime retour chariot automatique                |

| Nom        | abrév.               | défaut           | description                                                            |
|------------|----------------------|------------------|------------------------------------------------------------------------|
| prompt     | -                    | prompt           | invite : en mode ligne de cmd                                          |
| readonly   | ro                   | noro             | interdit la modification du fichier                                    |
| remap      | -                    | remap            | autorise macros dans des macros                                        |
| report     | -                    | report=10        | taille maxi. modifications notifiées                                   |
| scroll     | -                    | scroll=11        | nombre de lignes pour C-d et z                                         |
| shell      | sh                   | sh=\$SHELL       | nom du shell pour ! et :sh                                             |
| shiftwidth | sw                   | sw=8             | nombre d'espaces pour indenter                                         |
| showmatch  | sm                   | nosm             | indique le séparateur associé ([ ] { })                                |
| slowopen   | slow                 | slow             | retarde l'affichage lors des insertions                                |
| tabstop    | ts                   | ts=8             | taille des tabulations C-i                                             |
| taglength  | tl                   | tl=0             | taille minimum pour génération de tags ( 0 = pas de limite, v. ctags ) |
| tags       | "tags /usr/lib/tags" |                  | emplacement du fichier de tags                                         |
| term       | -                    | term=\$TERM      | nom du terminal utilisé                                                |
| terse      | -                    | noterse          | messages d'erreurs raccourcis                                          |
| timeout    | to                   | noto             | 1s maxi. pour taper une macro                                          |
| warn       | -                    | warn             | "changements non enregistrés"                                          |
| window     | -                    | window= <i>n</i> | nombre de lignes de la fenêtre texte                                   |
| wrapmargin | wm                   | wm=0             | retour automatique a <i>n</i> caractères                               |
|            |                      |                  | de la marge droite                                                     |
| wrapscan   | ws                   | ws               | les recherches à la fin du texte recommencent au début                 |
| writeany   | wa                   | nowa             | pas de vérification de droits d'accès                                  |

### Configuration au démarrage:

Les directives `:set` et `:map` d'un utilisateur peuvent être réunies dans le fichier `$HOME/.exrc`. Il ne faut pas de «:» dans ce cas.

On peut aussi les mettre dans la variable d'environnement EXINIT  
 ex : export EXINIT="set noautoindent aw showmatch" avec `bash`

## I n d e x

### Abréviations (Macros et abréviations)

Ajuster l'écran

Annuler et répéter

Appel de l'éditeur

Commandes 'shell'

Configurer l'éditeur

Démarrage (Appel de l'éditeur)

Démarrage (Configurer l'éditeur)

Déplacer le curseur

Etat du texte

Exécution d'une commande 'shell'

Filtrage par commande 'shell'

Insérer du texte (Modifier le texte)

Macros et abréviations

Manipuler des fichiers

Marquer le texte

Modes d'édition

Modifier le texte

Notations

Options (Configurer l'éditeur)

Rechercher et remplacer

Redessiner, recentrer l'écran (Ajuster l'écran)

UNIX™ est une marque déposée d' AT&T Bell Laboratories

© 1996 Logiciels du soleil