Vi

Qu'est-ce que vi ?

Dispersé dans tous votre ordinateur sont des centaines de fichiers texte. Pour un nouvel utilisateur, cela peut sembler sans importance, mais presque tout est géré par des fichiers de configuration en texte brut sous Slackware Linux. Cela permet aux utilisateurs d'effectuer des changements rapidement, facilement et intuitivement. Dans le chapitre 5 nous avons vu quelques commandes telles que ***cat*** et ***less*** qui peuvent être utilisées pour visualiser ces fichiers, mais que faire si nous voulons les modifier ? Pour cela nous avons besoin d'un éditeur de texte, et ***vi*** est parfait pour cela.

Pour faire court, ***vi*** est l'un des plus vieux et plus puissants éditeur de texte toujours en utilisation aujourd'hui. Il est chéri par des administrateurs systèmes, des programmeurs, des passionnés et autres de par le monde. En fait, ce livre tout entier à été écrit avec ***vi*** ; seul le chapitre concernant ***emacs*** à été écrit avec cet éditeur.

Une explication un peu plus complète est nécessaire pour comprendre exactement ce que ***vi*** est actuellement, techniquement Slackware Linux ne propose pas***vi***. Au lieu de cela Slackware propose deux "*clones*" de vi, ***elvis***(1) et ***vim***(1). Ces clones ajoutent de nombreuses fonctionnalités à vi telle que la coloration syntaxique, l'édition en mode binaire et la gestion du réseau. Nous n'irons pas dans tous les détails. Par défaut, si vous exécutez ***vi*** sur Slackware Linux, vous utiliserez ***elvis***, donc tous les exemples de ce chapitre assumeront que c'est ce que vous utiliserez. Si vous avez utilisé une autre distribution Linux auparavant, vous serez peut être un peu plus à l'aise avec ***vim***. Si c'est le cas, vous pourrez changer le lien symbolique `/usr/bin/vi` pour le faire pointer vers `/usr/bin/vim`, ou en ajoutant un alias dans les scripts de démarrage de votre shell. ***vim*** est souvent considéré comme plus riche en fonctions qu'***elvis***, mais ***elvis*** est un programme plus petit et comporte plus de fonctions que nécessaires pour la plupart des utilisateurs.

vi est très puissant, mais également un peu pénible et rude à apprendre pour un nouvel utilisateur. Toutefois, maîtriser ***vi*** est une compétence importante pour tout administrateur système qui se respecte, car ***vi*** est disponible dans quasiment toutes les distributions Linux, tous les systèmes BSD et tous les systèmes UNIX existants. Il est même proposé dans Mac OS X. Une fois que vous avez appris à utiliser ***vi***, vous n'avez plus besoin d'apprendre à utiliser un autre éditeur de texte pour travailler sur ces systèmes. En fait, des clones de ***vi*** ont été portés pour Microsoft Windows, vous pourrez l'utiliser sur ce système également.

Les différents modes de vi

Les nouveaux utilisateurs sont souvent frustrés lorsqu'ils découvrent ***vi*** pour la première fois. Lorsqu'il est lancé sans option ***vi*** affichera quelque chose comme ceci à l'écran.

```
~
~
~
~
~
```

```
~
~
~
~
~
~
                                                Command
```

À ce point, l'utilisateur commencera à taper et s'attendra à ce que les touches qu'il pressent apparaissent dans le document. À la place, des choses étranges se produiront. La raison en est simple. ***vi*** à différents "*modes*" d'opération. Il existe un mode de commande et un mode d'insertion. Le mode de commande est celui par défaut ; dans ce mode chaque frappe au clavier effectue une opération spécifique telle que déplacer le curseur, supprimer du texte, copier du texte, lancer une recherche, etc...

Ouvrir, sauvegarder et quitter

Vous avez donc décider d'apprendre à utiliser ***vi***. La première chose à faire est d'apprendre à ouvrir et enregistrer des fichiers. Ouvrir des fichiers est assez facile. Entrez simplement le nom du fichier comme option en ligne de commande et ***vi*** se fera une joie de l'ouvrir. Par exemple vi chapter_11.xml ouvrira le fichier `chapter_11.xml` et affichera son contenu à l'écran, tout simplement. Mais si nous avons fini avec un document et que nous souhaitons le sauvegarder ? Nous pouvons le faire en mode commande en utilisant la commande *:w*. Une fois en mode commande pressez la touche : qui positionnera temporairement le curseur tout en bas de la fenêtre et vous pourrez alors entrer des commandes spéciales (ceci est techniquement connu comme le mode ex d'après le vénérable programme ***ex*** que nous ne présenterons pas ici). La commande pour sauvegarder votre travail courant est *:w*. Une fois ceci effectué, ***vi*** enregistrera vos modifications depuis la mémoire tampon vers le fichier. Si vous souhaitez ouvrir un autre document, utilisez simplement la commande *:e autre_document* et ***vi*** se fera une joie de l'ouvrir pour vous. Si vous avez effectué des changements dans la mémoire tampon mais que vous n'avez pas encore sauvegardé *:e* échouera et affichera un message d'avertissement sur la dernière ligne. Vous pouvez outrepasser cela avec la commande *:e!*. La plupart des commandes en mode ex de ***vi*** peuvent être "*forcées*" en y ajoutant !. Cela indique à ***vi*** que vous voulez abandonner tous les changements que vous avez effectué dans le tampon et que vous voulez ouvrir immédiatement un autre document.

Mais si vous ne voulez pas garder vos modifications et que vous voulez quitter ou reprendre à zéro ? Cela peut aussi se faire facilement. Exécutez la commande *:e!* sans argument ré-ouvrira le document courant. Quitter ***vi*** se fait simplement avec la commande *:q* si vous n'avez effectué aucune modification ou *:q!* si vous souhaitez quitter sans les enregistrer.

Se déplacer

Se déplacer dans ***vi*** est peut être la chose la plus dure à apprendre pour un débutant. ***vi*** n'utilise pas les touches fléchées habituelles pour déplacer le curseur, encore que ce soit une option possible sous Slackware Linux. Au lieu de cela, les mouvements sont d'autres commandes utilisées en mode commande. La raison derrière cela est simple. ***vi*** est plus vieux que l'apparition des touches fléchées sur les claviers. De fait, les mouvements du curseur sont gérés en utilisant moins de touches que

celle disponibles et les touches de la ligne de base de la main droite[1] h, j, k et l furent choisies pour cela. Ces touches permettent de déplacer le curseur à partir du moment où ***vi*** est en mode commande. Voici un petit tableau pour vous aider à retenir comment elles fonctionnent :

Commande	**Résultat**
h	déplace le curseur d'un caractère vers la gauche
j	déplace le curseur d'une ligne vers le bas
k	déplace le curseur d'une ligne vers le haut
l	déplace le curseur d'un caractère vers la droite

Les déplacements sont en fait un peu plus puissant que cela. Tout commande d'autres touches de commandes, les touches de déplacement acceptent des arguments numériques. Par exemple **10j** déplacera le curseur de dix lignes vers le bas. Vous pouvez également atteindre la fin ou l'origine de la ligne courante avec les touches $ et ^ respectivement.

Éditer un document

Maintenant que nous sommes capables d'ouvrir et sauvegarder des documents, ainsi que de s'y déplacer, il est temps d'apprendre à les éditer. Le premier moyen pour l'édition est d'entrer en mode insertion en utilisant soit les touches de commandes i ou a. Elles permettent d'insérer du texte avant le curseur ou après le curseur, respectivement. Une fois en mode insertion vous pouvez taper votre texte normalement et il sera intégré dans votre document. Vous pouvez retourner dans le mode commande pour sauvegarder vos changements en appuyant sur la touche Échap.

Aide-mémoire vi

Comme ***vi*** peut être difficile à apprendre, j'ai préparé un petit aide mémoire qui pourra vous aider à débuter, en attendant d'être plus à l'aise.

Commande	**Résultat**
h	déplace le curseur d'un caractère vers la gauche
j	déplace le curseur d'une ligne vers le bas
k	déplace le curseur d'une ligne vers le haut
l	déplace le curseur d'un caractère vers la droite
10j	déplace le curseur de 10 lignes vers le bas
G	aller à la fin du fichier
^	aller au début de la ligne
$	aller à la fin de la ligne
dd	supprimer la ligne (et la copier en mémoire tampon)
5dd	supprimer 5 lignes (et les copier en mémoire tampon)
dw	supprimer un mot (et la copier en mémoire tampon)
5dw	supprimer 5 mots (et les copier en mémoire tampon)
yy	copier une ligne (et la copier en mémoire tampon)
yw	copier un mot (et le copier en mémoire tampon)
5yw	copier 5 mots (et les copier en mémoire tampon)

Commande	Résultat
p	coller le contenu de la mémoire tampon à l'emplacement du curseur
P	coller le contenu de la mémoire tampon au dessus de l'emplacement du curseur
r	remplace un caractère
R	remplace plusieurs caractères
x	supprime un caractère
X	supprime le caractère précédent
u	annule l'action précédente
:s'ancien'nouveau'g	remplace toutes les occurrences d'*ancien* par *nouveau* sur la ligne active
:%s'ancien'nouveau'g	remplace toutes les occurrences d'*ancien* par *nouveau* dans tous le document
/asdf	recherche la prochaine occurrence de *asdf*
:q	quitte (sans sauvegarder)
:w	enregistre le document courant
:w fichier	sauvegarde le document courant sous le nom *fichier*
:x	sauvegarde et quitte

Navigation

Chapitre précédent : Travailler avec les systèmes de fichiers

Chapitre suivant : Emacs

Sources

- Source originale : http://www.slackbook.org/beta

- Publication initiale d'Alan Hicks, Chris Lumens, David Cantrell, Logan Johnson
- Traduction initiale de escaflown
- Traduction de Ellendhel

slackbook, vi, text editor

1)

utilisées en dactylographie - NdT