

Présentation de «vi»

L'éditeur de texte «vi» ne fait pas partie des éditeurs les plus évolués et les plus conviviaux. Par contre c'est un standard sur tous les systèmes UNIX, c'est-à-dire que quelque soit la version et le fournisseur du système que vous utilisez, «vi» est implanté. Il est très puissant pour travailler sur des fichiers systèmes (application de filtres ...) et ne nécessite que très peu de ressources. En outre les directives de «vi» sont souvent utilisables dans des commandes UNIX (less, more, sed ...).

Commandes générales pour démarrer, quitter et annuler.

Lancement de l'éditeur (le nom est optionnel) **vi < nom_fichier >**

Sauvegarde (le nom est optionnel) **:w < nom_fichier >**

Sortie

sans modification et sans sauvegarde **:q**

après modification et sans sauvegarde **:q!**

avec sauvegarde **ZZ** ou **:x** ou **:wq**

Divers

annuler la dernière commande **u**

répéter la dernière commande **.**

exécuter n fois (par ex. 10) une commande (par ex. dd) **10dd**

réafficher l'écran **CTRL L**

appliquer un filtre (par ex. trier par ordre alphabétique les lignes de 5 à 10 dans le texte) **: 5 , 10 ! sort**

Directives

Une directive c'est un ou plusieurs caractères. Une fin de directive c'est soit une fin implicite, soit la touche **ESC**.

1. Directives avec fin implicite

\$ positionnement en fin de ligne

yy copie la ligne courante dans un tampon

nny copie n lignes dans un tampon

p copie le tampon après la ligne courante

x suppression du caractère courant

dw suppression du mot courant

dd suppression de la ligne courante

/chaîne recherche de l'occurrence d'une chaîne

2. Directives avec sortie par **ESC**

i insertion avant le curseur

a ajout après le curseur

Mode commande

On accède au mode commande (dernière ligne de l'écran) en tapant le caractère «:». Les commandes autorisées sont :

toutes les commandes de sauvegarde et de sortie vues précédemment

n1 : aller à la ligne n1

\$: aller à la dernière ligne

d : détruire la ligne courante

n1,n2d : supprimer les lignes n1 à n2

m n : déplacer la ligne courante en ligne n

n1,n2 m n3 : déplacer les lignes de n1 à n2 en ligne n3

t n : dupliquer la ligne courante en ligne n

n1,n2 t n3 : dupliquer les lignes de n1 à n2 en ligne n3

s/chaîne1/chaîne2/ : substitution dans la ligne courante de la première occurrence de chaîne1 par chaîne2

s/chaîne1/chaîne2/g : idem que précédemment mais sur toutes les occurrences de chaîne1

n1,n2s/chaîne1/chaîne2/g : idem pour les lignes de n1 à n2 ; ex. : 1,\$s/avoir/être/g

r : lire un fichier et l'insérer à la ligne courante

!commande : appeler une commande UNIX

Exercices

Placez vous dans une fenêtre sous le shell. Lancer l'éditeur « vi ». Après y avoir inséré le texte suivant, suivez les directives de modification du texte données ci-dessous :

Déplacer le deuxième paragraphe pour le mettre en fin de texte.

Commande : _____

Sauvegarder le contenu dans un fichier de nom « TDediteur » sans quitter « vi ».

Commande : _____

Insérer le texte suivant entre l'avant dernier et le dernier paragraphe : « Dans OS X El Capitan, les Apps vous offrent des moyens inédits et captivants de faire ce que vous faites le plus : surfer sur le Web, classer vos e-mails, prendre des notes ou encore retoucher des photos. ».

Commande : _____

Utiliser la commande de substitution pour remplacer partout dans le texte «OS X» par «Apple OS X».

Commande : _____

Ajouter en fin de fichier le contenu du fichier /etc/passwd.

Commande : _____

Utiliser la commande de substitution pour remplacer partout dans le texte «mac» par «Mac».

Commande : _____

Compter en utilisant les commandes de recherche de chaîne le nombre d'occurrence du mot « Apple».

Commande : _____

Appeler la commande « man vi » sans quitter l'éditeur.

Commande : _____

Quitter l'éditeur en sauvegardant le contenu.

Commande : _____

Vous voulez utiliser les directives de déplacement (par exemple « \$ ») mais le caractère \$ s'affiche dans votre texte. Pourquoi et que faire ?

Vous venez de détruire une ligne par erreur, que faire ?

Commande :