Université de Ouagadougou

Formation sur l'Administration des clusters bioinformatiques

Richard KOUTOUNGUIMINA

Chapitre I L'éditeur de texte vi

L'éditeur vi comprend 27 buffers dont 26 nommés correspondant aux 26 lettres de l'alphabet (a à z) et le 27 ième buffer non nommé a le même contenu que le buffer nommé dans lequel il y a eu la dernière copie.

L'éditeur vi dispose de 3 modes :

- le mode commande qui est le mode initial. Tout caractère tapé est interprété comme une commande
- le mode insertion qui permet de saisir du texte au clavier, le retour au mode commande se fait en appuyant sur la touche escape.
- Le mode dialogue qui permet de saisir une ligne de commande. Le mode dialogue est sélectionné soit pour exécuter une commande soit pour effectuer une recherche de chaine.

a) Déplacement du curseur

h caractère précédent (CTRL H)

l caractère suivant

j ligne suivante avec le curseur en même colonne (CTRL N)

k ligne précédente avec le curseur en même colonne (CTRL P)

+ début de ligne suivante

- Début de ligne précédente

0 début de ligne (marge gauche)

n| colonne n (par défaut 0)

^ début du premier mot de la ligne

$ fin de ligne

w mot suivant

b mot précédent

e fin de mot

CTRL F écran suivant

CTRL B écran précédent

CTRL D demi-écran suivant

CTRL U demi-écran précédent

CTRL E affichage d'une ligne supplémentaire en bas de l'écran

CTRL Y affichage d'une ligne supplémentaire en haut de l'écran

H positionnement en première ligne de l'écran

L positionnement en dernière ligne de l'écran

M positionnement sur la ligne médiane de l'écran

G positionnement sur la dernière ligne du fichier

nG positionnement sur la nième ligne du fichier (n entier)

À la place de h, l, j et k on peut utiliser les flèches.

b) Destructions

x destruction du caractère sous le curseur (idem dl)

X destruction du caractère précédent le curseur

dd destruction de la ligne courante

D destruction de la fin de la ligne courante à partir du curseur

d opérateur de suppression qui doit être suivi d'une position indiquant sur quelle portion porte la destruction :

db destruction du début du mot jusqu'au curseur exclus

dw destruction du curseur inclus jusqu'à la fin du mot.

d0 destruction du début de la ligne jusqu'au curseur

d$ destruction du curseur jusqu'à la fin de la ligne.

ndd destruction de n lignes à partir du curseur

nx destruction de n caractères à partir du caractère sous le curseur

nX destruction de n caractères précédents le caractère sous le curseur

c) Passage en mode insertion

i insertion avant le curseur

a ajout après le curseur

o ouverture d'une ligne après la ligne courante

O ouverture d'une ligne avant la ligne courante

I insertion en début de ligne

A insertion en fin de ligne

Touche ESCAPE sortie du mode insertion.

Pendant la saisie, le touche backspace est utilisée pour corriger le texte.

d) Remplacement de texte

rx remplacement du caractère sous le curseur par x

C remplacement à partir du curseur jusqu'à la fin de la ligne

S remplacement de la ligne courante

R remplacement à partir du curseur

c opérateur de remplacement qui agit come d c'est-à-dire qu'il faut le faire suivre d'une position (c$ équivaut à C, cc à S, cw au remplacement jusqu'à la fin du mot, etc).

Toutes ces commandes sauf rx font passer vi en mode insertion.

e) Déplacement et copie de texte

Un opérateur de copie (y) permet de copier une portion de texte dans un buffer (tampon). y doit être suivi d'une position pour délimiter le texte à copier, il peut être précédé d'un facteur multiplicateur. L'opérateur Y permet de faire une copie complète de la ligne entière (équivalent à yy).

Deux autres opérateurs (p et P) permettent d'insérer le texte d'un tampon soit avant le curseur (P) soit après le curseur (p). si le texte du tampon correspond à une ou plusieurs lignes, celles-ci sont insérées soit avant la ligne courante (P), soit après(p).

Par défaut ces opérateurs utilisent le tampon sans nom. Pour utiliser un autre tampon, on doit faire précéder les commandes de "x avec x le nom du buffer.

Exemples :

y$ copie de la fin de la ligne courante dans le buffer sans nom

P insertion du texte copié avant le curseur

"a3yy copie de 3 lignes dans le buffer a

"ap insertion du buffer a après la ligne courante.

Pour déplacer le texte, on utilise l'opérateur d à la place de y.

f) Autres commandes du mode dialogue

ZZ sauvegarde du texte sur le fichier courant et sortie de vi

U restauration de la ligne courante dans on état initial c'est-à-dire celui qu'elle avait lorsqu'on a positionné le curseur dessus.

u annulation de la commande précédente.

. répetition de la dernière commande de modification.

CTRL G abandon de la commande en cours.

n recherche de la prochaine occurrence (après une commande /chaine ou ?chaine)

N idem n mais avec recherche en arrière.

J concaténation de la ligne courante et de la suivante.

g) Commandes du mode dialogue

Toute commande du mode dialogue commence par :, /, ? et se termine par la touche return. Elle est affichée sur la dernière ligne de l'écran et peut être modifiée avec backespace.

:q sortie de vi

:wq sauvegarde sur le fichier courant et sortie de vi (équivalent à ZZ)

:w sauvegarde sur le fichier courant

:w f sauvegarde sur le fichier f

:n édition du fichier suivant si vi a été appelé avec plusieurs fichiers.

:sh appel au shell (retour par exit ou CTRL D).

:!cmd exécution de la commande cmd de UNIX

:!sh création d'un sous-shell, le retour sous vi s'effectue par CTRL D

:x exécution de la commande x

/ch recherche en avant de la chaine ch

?ch recherche en arrière de la chaine ch

:q! pour forcer la sortie sans sauvegarder

:w! pour forcer la sauvegarde quand le fichier est en lecture seule.

:wq! pour forcer la sauvegarde sur le fichier courant et la sortie de vi.

:wq! f pour forcer la sauvegarde sur le fichier f et la sortie de vi.

Chapitre II : Enchainement des commandes-redirections-tubes

I) Enchainement des commandes

Il y deux types d'enchainement : l'exécution séquentielle et l'exécution conditionnelle.

- L'exécution séquentielle : les commandes sont séparées par un ;
- L'exécution conditionnelle, les commandes sont séparées par && ou || :
    - && la commande suivante n'est exécutée que si celle qui la précède l'a été.
    - || la commande suivante est exécutée quand celle qui la précède a échoué.

Exemples :

- who ;date ;pwd
- mv f1, f2 && pwd
- mv f1 f2 || pwd

II) Redirections

L'utilisateur qui se connecte dispose de 3 fichiers ouverts :

- L'entrée standard (notée stdin :0) affectée au clavier
- La sortie standard (notée stdout :1) affectée à l'écran
- La sortie des erreurs (notée stderr :2) affectée à l'écran.

Il est possible de modifier ces affectations par défaut le temps d'une commande : on parle de rédirection.

1) Redirection de la sortie

>Redirige la sortie dans le fichier. Si le fichier n'existe pas il est créé, s'il existe, son contenu est écrasé.

Syntaxe : cmd > nom_fichier

>> redirige la sortie à la suite du contenu du fichier mentionné.

Syntaxe : cmd >> nom_fichier

On peut rediriger la sortie et les erreurs en même temps :

Syntaxes :

- Cmd >& nom_fichier redirige la sortie et les erreurs dans le fichier mentionné.
- Cmd >>& nom_fichier redirige la sortie et les erreurs à la suite du contenu du fichier mentionné.

2) Redirection de l'entrée

Pour rediriger l'entrée on utilise le caractère < suivi d'un nom de fichier existant.

Syntaxe : cmd < nom_fichier

Exemple : write richard < /etc/shells

richard étant un login d'un utilisateur connecté au système UNIX.

III) Les tubes

Certains shells tels que le C-shell et le bash permettent de rediriger la sortie d'une commande sur l'entrée d'une deuxième commande : on parle de tube ou pipe. On utilise le caractère | entre les commandes.

Syntaxe : cmd1 | cmd2 ou cmd1| cmd2|cmd3 …..

Il est possible de combiner les tubes, les redirections ou d'enchainer plusieurs tubes.

Exemple : who| wc -l