

Contexte d'une session de travail

1. Interpréteur de commandes

La connexion réussie d'un utilisateur (ici le `root` car on se base sur une version serveur) a activé l'interpréteur de commandes approprié (indiqué dans le fichier `/etc/passwd`). L'interpréteur de commandes (ou `shell`) représente l'interface entre l'utilisateur et le système d'exploitation. Comme toujours, et Ubuntu ne déroge pas à la règle, il existe non pas un mais plusieurs interpréteurs de commandes. Le `shell` par défaut sous Ubuntu est le BASH ou *Bourne Again Shell*.

Pour expliquer les notions qui suivent, la commande basique (et essentielle) `ls` qui liste les fichiers d'un répertoire sera utilisée.

a. Englobement ou expressions rationnelles du shell

L'englobement dans une commande du shell consiste à taper un ou plusieurs caractères spéciaux signifiant une sélection particulière à appliquer à cette commande. Caractère spécial le plus courant : le caractère "étoile" qui remplace une suite de caractères quelconques :

```
ls b*e
```

Cela affiche la liste de tous les fichiers du répertoire courant commençant par un `b` et finissant par un `e`.

```
ls /etc/*.conf
```

Liste tous les fichiers du répertoire `/etc` ayant pour extension `.conf`.

Autres caractères d'englobement

`?` : remplace un seul caractère

`[...]` : correspond aux caractères entre crochets soit :

`[1-9]` : pour des chiffres

`[GER]` : pour un G, un E ou un R

`[a-z]` : pour tout l'alphabet en minuscule

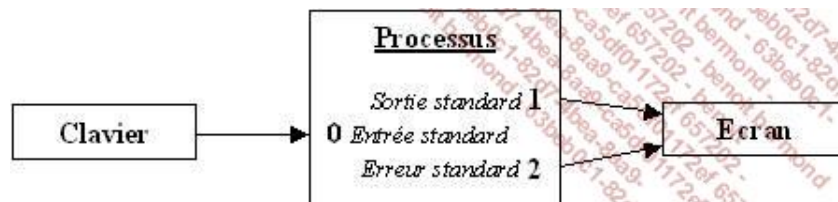
`[!...]` : même chose mais en exclusion

```
ls b[!0-9]e
```

Cette commande liste les fichiers commençant par un `b`, avec une deuxième lettre qui ne peut être un chiffre et se terminant par un `e`.

b. Redirections

Le système Linux Ubuntu comporte une entrée dite standard réservée au clavier par défaut et une sortie standard consacrée à l'écran. Pour compléter cet équilibre, une sortie d'erreur standard (toujours l'écran par défaut) est ajoutée. Pour le shell, les codes associés à ces flux correspondent respectivement à 0, 1 et 2.



Le principe de redirection consiste redéfinir ces flux standards. L'utilisateur réalise ce type d'opérations constamment sans le savoir lorsqu'il imprime un document. La sortie est redirigée vers l'imprimante et non plus vers l'écran. Les redirections possibles sont :

- un **fichier** : utile pour enregistrer des informations ou des événements comme les erreurs.
- un **tube** : on redirige la sortie de la commande vers l'entrée d'une autre (voir le détail plus bas).
- un **périphérique** : classique comme l'imprimante.

La redirection utilise le caractère > pour fonctionner, précédé ou non (implicite pour le 0 ou 1) du code de flux. Un double >> détermine un ajout pour un fichier alors que le caractère > simple détruit le fichier si existant avant de le remplacer par le contenu du flux.

En voici une syntaxe plus précise :

```
0<nom_de_fichier
    nom_de_fichier comme entrée standard, code 0 optionnel
1>nom_de_fichier
    nom_de_fichier comme sortie standard, code 1 optionnel
1>>nom_de_fichier
    idem en ajout au fichier et non en création, code 1 optionnel
2>erreurs
    dirige les erreurs vers le fichier erreurs
```

Exemples :

```
ls /etc/*.conf > liste.txt
```

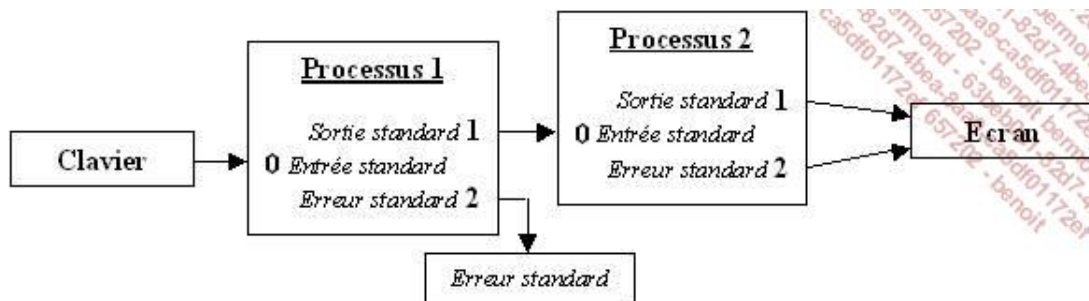
```
ls /usr/local/etc/*.conf >> liste.txt
```

```
ls /home/max/ > liste_max.txt 2> erreurs.txt
```

Le premier exemple envoie la liste des fichiers du répertoire /etc ayant l'extension .conf dans le fichier liste.txt (il n'y a plus d'affichage à l'écran). Le deuxième exemple ajoute à ce fichier la liste des fichiers de même extension trouvée dans /usr/local/etc. Le dernier exemple envoie la liste des fichiers du répertoire max dans le fichier liste_max.txt et un message d'erreur dans le fichier erreurs.txt si l'utilisateur max n'existe pas (dans ce cas, le fichier liste_max.txt est vide).

c. Tubes

Un tube combine une redirection d'entrée et de sortie, ou plus exactement relie la sortie standard d'une commande à l'entrée standard d'une autre commande. Ceci se matérialise dans la ligne de commande par les deux tirets verticaux | (touche [Alt Gr] 6).



Cette technique s'emploie couramment dans les tâches administratives. Des exemples plus concrets sont décrits dans les chapitres sur la maintenance. Avec la commande `ls`, on utilise typiquement un tube avec la commande `less`.

Cette commande découpe en page la sortie à l'écran pour les fichiers très longs qui, sans cela, défileraient jusqu'à la fin. Le déplacement se réalise à l'aide des touches [Page Up] et [Page Down] (les flèches du pavé numérique pour les pages) car on peut remonter dans le fichier à l'inverse d'une autre commande similaire, la commande `more`.

Exemple :

```
ls -l /etc/ | less
```

L'affichage du répertoire `/etc/` dépasse la capacité de l'écran. L'affichage entier est passé en entrée de la commande `less` et fractionné. Pour sortir à tout moment du tube, on tape la lettre `q`.

➤ Un truc très utile pour les administrateurs : même si vous n'avez pas fait de tube, l'utilisation des touches [Page Up] et [Page Down] permettent de revenir en arrière et en avant de l'affichage. La seule limite réside dans la taille du tampon (buffer) de la sortie écran.

2. VIM ou l'éditeur de l'administrateur

a. Éditeur et traitement de texte

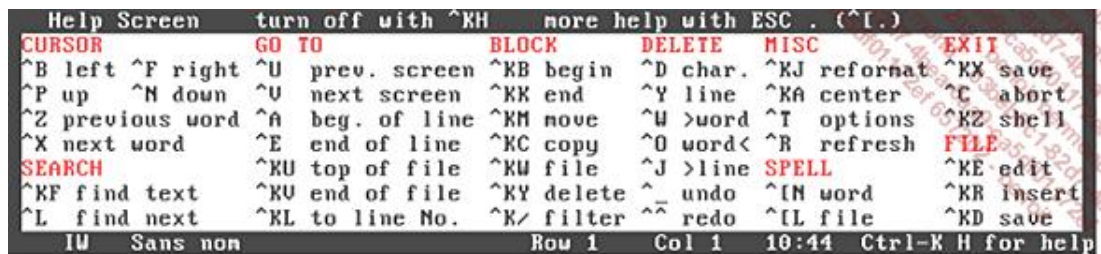
Un éditeur de texte sert à créer et/ou modifier des fichiers texte sans mise en forme. Une simple mise en page pour la lisibilité suffit à la différence d'un traitement de texte qui peut utiliser outre une présentation élaborée, des objets comme des images, des graphiques, etc. On peut dissocier donc l'écriture (faite par un éditeur de texte) de la mise en page définitive (faite par un compilateur de texte comme **LaTeX** par exemple). Dans le cas d'un traitement de texte, ces deux étapes sont simultanées.

Comment choisir son éditeur ? Certains sont très simples, voire spartiates, d'autres simples mais graphiques et avec une souris, d'autres plus complexes... Le choix est simple car fonction de l'environnement :

Pour la console texte

- `vi` (ou en fait `vim`)
- `pico` (ou `nano` déjà utilisé dans le précédent chapitre)
- `joe`

Tous ces éditeurs possèdent leurs raccourcis de commandes et, il faut le dire, souvent abscons. Voici par exemple les commandes de l'éditeur `joe` :



Help Screen turn off with ^KH more help with ESC . (^[.)					
CURSOR	GO TO	BLOCK	DELETE	MISC	EXIT
^B left ^F right	^U prev. screen	^KB begin	^D char.	^KJ reformat	^KX save
^P up ^N down	^V next screen	^KK end	^Y line	^KA center	^C abort
^Z previous word	^A beg. of line	^KM move	^W >word	^T options	^KZ shell
^X next word	^E end of line	^KC copy	^O word<	^R refresh	FILE
SEARCH	^KU top of file	^KW file	^J >line	SPELL	^KE edit
^KF find text	^KV end of file	^KY delete	^_ undo	^IN word	^KR insert
^L find next	^KL to line No.	^K/ filter	^^ redo	^IL file	^KD save
IW Sans non		Row 1	Col 1	10:44	Ctrl-K H for help

Pour l'environnement graphique (liste non exhaustive)

- `gedit` (environnement GNOME)
- `kate` (environnement KDE)
- `gvim` (version graphique de `vim`)
- `xemacs` (version graphique d'`emacs`)

b. L'éditeur de texte VI


Le principe d'utilisation d'un éditeur de texte pour un administrateur se rapporte essentiellement à sa rapidité. L'administrateur système ne programme pas, il crée certes des scripts (courts) mais son activité essentielle avec un éditeur de texte réside dans l'édition de fichiers de logs, (journaux), dans l'édition et la modification de fichiers de configuration de services.

Pourquoi VI (on prononce vi-aïe) ou plutôt VIM (**VI** **IM**proved soit **VI** amélioré) ? Parce qu'il fait partie intégrante de tout environnement Unix/Linux et qu'il représente un outil de base incontournable : quelle que soit la distribution, quel que soit le système, il existe une version de VI. Son apprentissage fait partie de la fondation même de la fonction d'administrateur. Ne pas connaître VIM et se prétendre administrateur système fera immédiatement naître une suspicion vis-à-vis de vos capacités réelles dans la fonction.

Lancement de VIM

L'installation de la distribution Ubuntu, qu'elle soit serveur ou poste de travail, propose une version de base de VIM (vim-common et vim-tiny). Dans cette version, certaines manipulations ne sont pas possibles comme par exemple l'utilisation des touches de curseur pour naviguer dans le texte. Le gestionnaire de paquets doit normalement être correctement configuré pour pouvoir installer le paquetage VIM afin d'obtenir la version complète :

```
aptitude install vim
```

 L'utilisation de l'éditeur VIM permet de mettre en lumière un comportement de Linux : la notion d'alternatives. En fonction de l'installation des logiciels, une même commande aboutit au lancement d'un logiciel différent. Sous Ubuntu, ce changement de comportement est visible dans le dossier /etc/alternatives. Dans le cas de VIM, avant l'installation du logiciel, les commandes vi ou vim suivent le lien /usr/bin/vim.tiny ; après l'installation, le lien se transforme en /usr/bin/vim.basic. Cette situation se retrouve dans l'exploitation d'un service de messagerie ou le choix du logiciel serveur n'influence pas les commandes de gestion du courrier. Si l'on revient dans le domaine des éditeurs de texte, les commandes pico, editor et nano aboutissent sous Ubuntu à ce même éditeur nano.

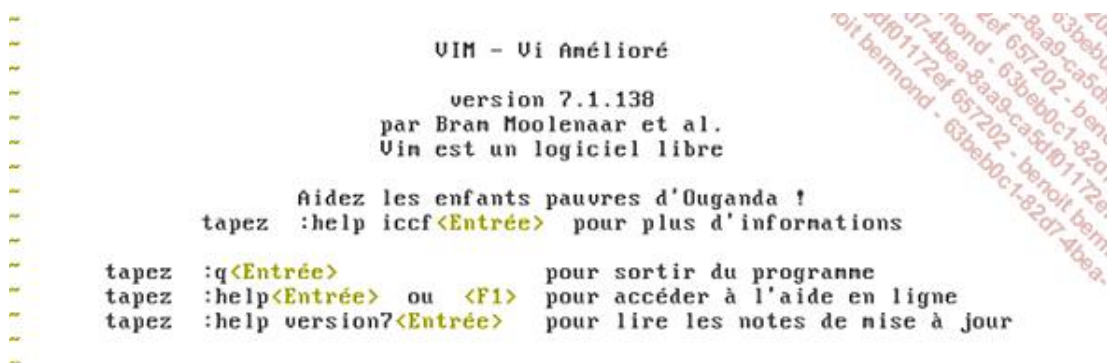
Une fois l'installation effectuée, la syntaxe générale de lancement de VIM est la suivante :


```
vi(m) [options] [chemin/nom_de_fichier]
```

En fonction ou non de la présence d'un nom de fichier à la suite de la commande, on aura :

- Si le fichier existe, l'éditeur l'ouvre en modification.
- Si le fichier n'existe pas, l'éditeur l'ouvre en création.
- Si aucun nom de fichier, la demande se fera à la sortie de l'éditeur.
- Pour être sûr d'ouvrir un fichier uniquement en lecture, on utilise l'option -R.

Voici l'écran d'accueil de VIM, l'écran s'efface et la colonne la plus à gauche affiche des ~ (tildes) :



 Notez l'appel humanitaire pour les enfants du centre ougandais de Kibaale. Cela a toujours été la règle pour VI et cela montre une fois de plus que logiciel libre rime avec esprit d'entraide communautaire.

Coloration syntaxique

L'éditeur VI possède la capacité de la coloration syntaxique en fonction du langage utilisé (BASH ou Perl, les plus utilisés par l'administrateur). Pour activer cette fonctionnalité, il suffit de modifier le fichier de configuration de VI :

- Éditez par VI le fichier /etc/vim/vimrc et supprimez le guillemet (") de commentaire sur la ligne syntax on.

- Lancez par exemple le fichier `.bashrc` dans votre répertoire de travail pour en voir le résultat.

c. Utilisation optimisée de VIM

Il existe beaucoup de littérature sur VI décrivant en détail et en totalité les commandes de l'éditeur. Pourtant même si VI constitue le premier élément de la trousse à outils de l'administrateur, il n'en reste pas moins vrai qu'il ne déroge pas à la règle classique qui consiste à ce qu'un utilisateur ne connaisse et ne tire profit en général que d'une partie d'un logiciel. Concrètement, l'apprentissage de VI est difficile et rébarbatif... Mais une fois maîtrisé, l'outil se révèle **indispensable**.

La présentation qui suit n'est pas exhaustive mais constitue la base de l'utilisation de VI tel qu'un administrateur le pratique au jour le jour. En clair : toutes les commandes ne sont pas décrites, mais l'essentiel est là.



Les débutants désireux de suivre une progression tireront profit du tutoriel fourni avec l'éditeur. Pour le lancer, soyez dans une console texte bien sûr et tapez la commande `vimtutor` :

```
=====
= Bienvenue dans le Tutoriel de VIM - Version 1.5.fr.2 =
=====

Vim est un éditeur très puissant qui a trop de commandes pour pouvoir
toutes les expliquer dans un cours comme celui-ci, qui est conçu pour en
décrire suffisamment afin de vous permettre d'utiliser simplement Vim.

Le temps requis pour suivre ce cours est d'environ 25 à 30 minutes, selon
le temps que vous passerez à expérimenter. Les commandes utilisées dans
les leçons modifieront le texte. Faites une copie de ce fichier afin de
vous entraîner dessus (si vous avez lancé "vimtutor" ceci est déjà une
copie).

Il est important de garder en tête que ce cours est conçu pour apprendre
par la pratique. Cela signifie que vous devez exécuter les commandes
pour les apprendre correctement. Si vous vous contentez de lire le
texte, vous oublierez les commandes !

Maintenant, vérifiez que votre clavier n'est PAS verouillé en majuscules,
et appuyez la touche j le nombre de fois suffisant pour que la leçon
1.1 remplisse complètement l'écran.
=====
Leçon 1.1 : DÉPLACEMENT DU CURSEUR
"/tmp/tutorXP4066" [convertit 809 lignes, 30059 caractères]
```

Les états de VI

On distingue sous VI deux états principaux :

- **mode commande** : saisie d'une commande.
- **mode saisie** : saisie d'un texte dans le buffer.

Au démarrage, VI se positionne automatiquement en mode commande. L'utilisation de VI implique que l'on passe fréquemment d'un mode à l'autre. En fait pour savoir où l'on est, il suffit de taper deux fois sur la touche [Echap] : vous êtes alors sûr d'être en mode commande.

Pour passer d'un mode à l'autre :

- du mode commande en mode saisie : appui d'une commande codée sur une seule lettre.
- du mode saisie en mode commande : appui de la touche [Echap] (bip si vous y êtes déjà).

Les autres modes de fonctionnement

Le mode `ex`, qui traite en partie de la façon de sortir de VI (des combinaisons sont possibles) :

Fonctions	Touches
-----------	---------

normalement sans sauvegarde	:q
obligatoirement sans sauvegarde	:q!
avec sauvegarde	:wq ou :x
avec sauvegarde mais sans sortie	:w
avec sauvegarde, sortie et indication du fichier	:wq <i>nom_de_fichier</i>
pour le fichier d'aide	:h (sortie par :q)

VI possède aussi la possibilité de travailler avec plusieurs fichiers à l'écran, mais nous ne le verrons pas. C'est un choix personnel : si l'on commence à utiliser VI comme un traitement de texte, alors autant en utiliser un et graphique de préférence ! VI doit rester ce qu'il est : un éditeur en mode console, simple et surtout rapide à utiliser.

Les commandes essentielles

Rappel : si vous n'êtes pas sûr de l'être, tapez deux fois sur [Echap], vous serez en mode commande. Un bip vous le signale. Vous avez, en mode saisie ou commande, la possibilité de vous déplacer avec toutes les flèches de direction, sauts de page, [Suppr], [Inser], etc.

En mode commande :

Fonctions	Touches
Insère du texte devant le curseur	i
efface et supprime la ligne entière (variante : <i>ndd</i> supprime <i>n</i> lignes)	dd 3dd (pour 3 lignes)
cherche le motif toto dans le texte à partir de curseur	/toto
idem sauf que la recherche se fait en arrière, vers le début du texte	?toto
répète la dernière commande de recherche	n
copie la ligne dans le Presse-papiers (variante : <i>nyy</i> copie <i>n</i> lignes)	yy 3yy (pour 3 lignes)
colle le contenu du Presse-papiers derrière le curseur	p
annule la dernière commande	u
répète la dernière commande	.

En mode ex :

Fonctions	Touches
aller à la ligne <i>n</i>	:n
aller à la dernière ligne	:\$
déplacer les lignes de <i>n1</i> à <i>n2</i> en <i>n3</i> (:2,4 m 8 déplace les lignes 2 à 4 après la ligne 8)	: <i>n1</i> , <i>n2</i> m <i>n3</i>
remplace chaine1 par chaine2 de <i>n1</i> à <i>n2</i> (si <i>n2</i> =\$ alors fin de	: <i>n1</i> , <i>n2</i> s/ <i>chaine1</i> / <i>chaine2</i>

fichier)	
insère un fichier à la ligne courante	:r <i>nom_de_fichier</i>
afficher les numéros de ligne (nonu pour l'enlever)	:set nu



Toutes les éditions de fichiers dans l'ouvrage sont faites par l'éditeur VIM.
