

## TP N°3 : Au cœur des fichiers

### L'éditeur vi

Vi est un éditeur de texte installé par défaut par Ubuntu. A la différence du Bloc-notes, ou des autres éditeurs de textes connus, **vi** est un éditeur de texte en ligne de commandes. C'est-à-dire que vous allez oublier la souris et que toutes les actions habituelles en traitement de texte : sauvegarde, copie...devront se faire à la main, via des commandes.

Il faut savoir que dans les entreprises, les systèmes d'exploitation UNIX installés sur les serveurs sont accessibles en ligne de commandes uniquement, via des terminaux. Ainsi **vi** sera le seul éditeur de texte que vous pourrez utiliser pour la création de vos scripts. Profitez-donc pleinement de ce TP !

### Les commandes vi

Vous pouvez retrouver toutes les commandes de l'éditeur **vi** sur ce lien :

[https://doc.fedora-fr.org/wiki/Utilisation\\_de\\_vi](https://doc.fedora-fr.org/wiki/Utilisation_de_vi)

Dans ce TP, nous ne présenterons que les commandes les plus importantes. Mais avant de parler commandes, il faut savoir que l'éditeur **vi** fonctionne suivant deux modes :

- **Mode édition** : c'est dans ce mode qu'on peut saisir du texte. Il faut taper la touche [ESC] pour passer en mode commande
- **Mode commande** : c'est dans ce mode qu'on peut déplacer le curseur, supprimer une ligne, sauvegarder le fichier...etc. Pour passer en mode édition, il faut utiliser une des commandes d'édition ci-dessous.

Insérer du texte (mode insertion)	<b>i</b>	active le mode insertion
	<b>a</b>	active le mode insertion, un caractère après le curseur
	<b>o</b>	insère une ligne en dessous du curseur et passe en mode insertion
Sauvegarder le fichier (mode commande)	<b>:x</b>	sauvegarde puis quitte <b>vi</b>
	<b>:q!</b>	quitter vi sans sauvegarder
Supprimer du texte (mode commande)	<b>x</b>	supprime un caractère
	<b>dd</b>	supprime une ligne
Copier-coller du texte (mode commande)	<b>y</b>	copie une ligne
	<b>p</b>	colle la ligne après le curseur
Divers (mode commande)	<b>u</b>	annule la dernière modification
	<b>/mot</b>	cherche « mot » en allant vers la fin du fichier

### Votre premier texte

Commencer par ouvrir un terminal. Taper la commande suivante : **vi test**

Attention : le fichier « test » ne sera effectivement créé que si vous le sauvegardez. Si le fichier « test » existe déjà son contenu ne sera pas écrasé.

A l'aide de l'éditeur **vi** écrire le texte suivant :

```
Un nouveau jour se lève
Je suis motivé
Rien ne m'arrêtera
Je poursuivrai mon rêve
Jusqu'au bout
Petit à petit
Pas à pas
Pas à pas
Demain commence maintenant
Demain commence maintenant
```

Une fois le texte saisi, sauvegarder et quitter l'éditeur **vi**.

### Quelques commandes UNIX pour traiter le fichier « test »

**cat** : affiche le contenu d'un fichier

Syntaxe : **cat** [**option**] [**chemin d'accès**]**nom\_fichier**

L'option **-n** affiche le fichier en numérotant les lignes à partir de 1.

**more** : affiche le contenu d'un fichier page par page

Syntaxe : **more** [**chemin d'accès**]**nom\_fichier**

**wc** : affiche le nombre de lignes, de mots ou de caractères contenus dans un fichier

Syntaxe : **wc** [**option**] [**chemin d'accès**]**nom\_fichier**

L'option **-l** affiche le nombre de lignes

L'option **-w** affiche le nombre de mots

L'option **-c** affiche le nombre de caractères

Application : tester la commande **wc** pour le fichier « test » avec une option à la fois. Puis essayer des combinaisons des options.

**tail** : affiche les dernières lignes d'un fichier

Syntaxe : **tail** [**-n**] [**chemin d'accès**]**nom\_fichier**

**N.B** : l'option **-n** permet d'afficher les **n** dernières lignes d'un fichier. Si **n** n'est pas précisé, les 10 dernières lignes sont affichées.

**head** : affiche les premières lignes d'un fichier

Syntaxe : **head** [**-n**] [**chemin d'accès**]**nom\_fichier**

**N.B** : l'option **-n** permet d'afficher les **n** premières lignes d'un fichier. Si **n** n'est pas précisé, les 10 premières lignes sont affichées.

**sort** : trie les lignes d'un fichier

Syntaxe : **sort** [**option**] [**chemin d'accès**]**nom\_fichier**

**N.B** : l'option **-r** permet d'inverser l'ordre du tri.

**grep** : cherche une chaîne de caractères dans un ou plusieurs fichiers

Syntaxe : **grep [option] chaîne [chemin d'accès]nom\_fichier**

L'option **-v** affiche les lignes ne correspondant pas à chaîne.

L'option **-n** affiche le numéro des lignes correspondantes.

L'option **-i** pour ignorer la casse.

Application : que va afficher la commande : **grep -vi demain test**

**find** : cherche un fichier dans un répertoire, ainsi que dans sa descendance (ses sous-répertoires).

Syntaxe : **find chemin option nom\_fichier**

L'option **-name** cherche le fichier par son nom.

L'option **-iname** même chose que **-name** mais insensible à la casse.

L'option **-mtime** cherche par date de dernière modification.

Exemple : **find . -mtime -2 -name "\*.conf"**

Cette commande cherche les fichiers du répertoire courant qui ont été modifiés entre maintenant et il y a 2 jours et dont l'extension est « .conf ».