

# Gestion des fichiers – Editeur vim

F. Lassabe

Printemps 2021

## Contents

<b>1 Liens symboliques et physiques</b>	<b>1</b>
<b>2 Gestion de l'espace de travail</b>	<b>2</b>
<b>3 Éditeur de texte <i>vim</i></b>	<b>4</b>
3.1 Tutoriel . . . . .	4
3.2 Exercice avec <i>vim</i> . . . . .	4
<b>A Rappel des commandes et modes de <i>vim</i></b>	<b>5</b>
A.1 Mode de <i>vim</i> . . . . .	5
A.2 Interface et commandes . . . . .	5

## 1 Liens symboliques et physiques

Rappel sur la visualisation et la création des liens :

**Afficher les inodes** `ls -li`

**Créer un lien symbolique** `ln -s fichier-réel fichier-lien`

**Créer un lien physique** `ln fichier-réel fichier-lien`

Répondez aux questions suivantes :

1. Créer le fichier texte `fichier.txt` en utilisant l'éditeur `gedit` (commande `gedit fichier.txt`), puis saisir trois à quatre lignes dans le fichier.
2. Quel est l'inode de ce fichier ?
3. Créer un lien symbolique `s_fichier.txt` et un lien physique `ph_fichier.txt` vers `fichier.txt`.
4. Afficher les inodes des trois fichiers. Que remarquez-vous ?
5. Ouvrir le fichier `fichier.txt` avec `gedit` et ajouter quelques lignes.
6. Ouvrir les fichiers liens. Est-ce que les contenus sont synchronisés ?

7. Refaire les étapes 5 et 6 en modifiant dans un premier temps le fichier `s_fichier.txt` puis le fichier `ph_fichier.txt`. Que se passe-t-il ?
8. Supprimer le fichier `fichier.txt` avec la commande `rm` (`rm fichier.txt`).
9. Vérifier les i-nodes des deux liens. Que remarquez-vous ?
10. Essayer de créer des liens physiques et symboliques sur le répertoire `Images` de votre répertoire personnel. Que se passe-t-il ?

## 2 Gestion de l'espace de travail

Dans cet exercice, vous allez manipuler des fichiers et des répertoires en utilisant les commandes de manipulation de fichiers suivantes :

**cat** permet de lister le contenu d'un fichier et de rendre la main au shell

**more** permet de lister le contenu d'un fichier avec possibilité de naviguer dedans

**less** permet de lister le contenu d'un fichier avec possibilité d'y naviguer. `less` est plus complet que `more`, et *less is more* (il permet de faire tout ce que fait `more`, et plus)

**cp** permet la copie de fichiers

**mv** permet de déplacer/renommer des fichiers (et répertoires)

**rm** permet de supprimer des fichiers (et des répertoires)

**touch** permet de créer ou de mettre à jour la date de modification d'un fichier

**vi** permet d'éditer des fichiers texte dans la console

**vim** comme `vi`, en plus pratique

Et les commandes de manipulations de répertoires suivantes :

**pwd** affiche le répertoire courant du shell exécuté

**cd** se déplace dans les répertoires

**mkdir** crée un répertoire

**rmdir** supprime un répertoire (vide)

Pour toutes les commandes shell, la commande `man <commande>` permet d'afficher un manuel d'utilisation de la commande *commande*. Il est même possible d'afficher le manuel de la commande *man* par la commande `man man`.

1. Afficher le manuel des commandes ci-dessus avec la commande `man`. Ne pas tout lire, c'est bien trop long. La structure est toujours plus ou moins la même :

- Un synopsis (résumé de la commande et des formes les plus courantes utilisées)
  - Liste des options possibles et de leur effet
  - Informations sur les auteurs de la commande
  - Exemples
2. Visualiser le contenu du fichier `/etc/services` avec `cat`, puis avec `more`. Quelle différence pouvez-vous remarquer entre ces deux commandes ?
  3. Créer l'arborescence illustrée par la figure 1, où *Répertoire d'accueil* désigne votre répertoire personnel (où vous arrivez grâce à une des commandes suivantes : `cd`, `cd $HOME` ou `cd ~`).
  4. Copier le fichier `/etc/fstab` dans *Rep3* sous le nom `fstab.txt`
  5. Renommer le fichier `fstab.txt` en `fstab.bis.txt` en le déplaçant vers *Rep4*.
  6. Lire le descriptif de la commande `wc` en utilisant `man`. A quoi sert-elle ?
  7. Tester cette commande avec ses différentes options sur le fichier `fstab.bis.txt`.
  8. Supprimer dans l'ordre le répertoire *Rep1* puis *Rep4* puis *Rep2*. Comment avez-vous fait ?

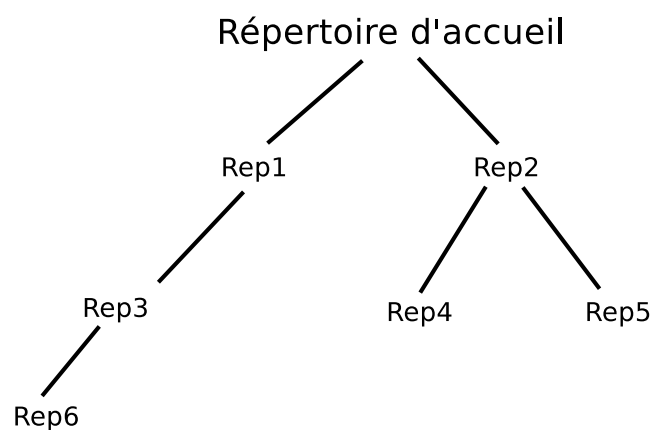


Figure 1: Arborescence à créer.

## 3 Éditeur de texte *vim*

L'éditeur *vim* est un éditeur de texte modal, fonctionnant dans un terminal. Bien qu'il soit moins ergonomique que les éditeurs courants (gedit, notepad, etc.), il est très puissant grâce aux plugins existant, et le fait qu'il fonctionne en mode texte dans une console le rend très utile (et utilisé) dans toutes les tâches d'administration distante de serveurs sans interface graphique installée.

### 3.1 Tutoriel

Lancer et suivre le tutoriel *vim* grâce à la commande `vimtutor`.

### 3.2 Exercice avec *vim*

Quand le tutoriel est terminé, réaliser la tâche suivante **avec vim** :

1. Créer le fichier *TP1\_essai* dans votre répertoire personnel.
2. Taper vos nom, prénoms, et adresse, chaque information sur une ligne différente.
3. Enregistrer le fichier.
4. Insérer une ligne après le nom, y écrire votre numéro de téléphone.
5. Effacer le prénom.
6. Insérer quelques lignes avec vos diplômes antérieurs (bac et brevet) et vos établissements fréquentés.
7. Déplacer les 3 premières lignes à la fin du fichier.
8. Insérer "Monsieur" ou "Madame" avant votre nom.
9. Remplacer un caractère accentué par le caractère non accentué correspondant.
10. Enregistrer ces modifications dans le fichier *TP1\_essai2*.

# Annexes

## A Rappel des commandes et modes de *vim*

### A.1 Mode de *vim*

Comme l'illustre la figure 2, *vim* comporte 3 modes de fonctionnement :

**le mode interactif** c'est le mode par défaut (en lançant *vim*). C'est un mode dans lequel il n'est pas possible d'écrire du texte, mais qui permet de manipuler le texte déjà présent (copier, coller, se déplacer, annuler des actions, etc.). Chaque action différente est déclenchée par l'appui sur une touche (dont la lettre est souvent relative à l'action, comme **P** pour **Paste**, c'est à dire **Coller** en Français).

**le mode insertion** c'est le mode de saisie du texte. On en sort avec la touche **Échap** qui nous remet alors en mode interactif.

**le mode commande** il permet de lancer des commandes (saisies dans la ligne du bas du terminal), comme sauver le fichier, chercher, chercher et remplacer, quitter, etc. On entre dans ce mode avec la touche ":", et on en sort soit avec la touche **Échap**, soit en validant la commande par l'appui sur la touche **Entrée**. On revient alors en mode interactif.

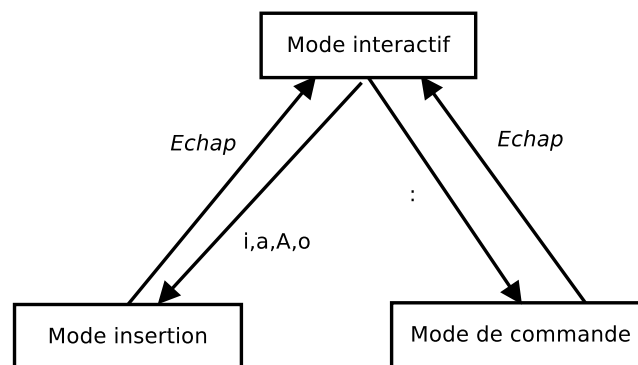


Figure 2: Modes de *vim* et méthodes de changement de mode.

### A.2 Interface et commandes

Ci-dessous, quelques commandes et touches de l'interface de *vim* sont décrites. Vous y retrouverez certaines actions et commandes du *vimtutor*.

Mouvements	Description
\$	Aller à la fin de la ligne
0 (zéro)	Aller au début de la ligne
gg	Aller au début du document
G	Aller à la fin du document
:n	aller à la ligne <i>n</i> ( <i>n</i> est le numéro de ligne souhaitée)
Commande	Effet
i	Entre en mode insertion au niveau du curseur
a	Entre en mode insertion après le curseur
A	Entre en mode insertion après le dernier caractère
o	Insère une ligne vide sous la ligne du curseur, place le curseur sur cette nouvelle ligne, et entre en mode insertion
O	Comme "o" mais sur la ligne du dessus
R	Entre en mode d'insertion par remplacement (écrase le texte sous le curseur au lieu d'insérer à sa position)
Échap	Revient au mode interactif
:	Entre dans le mode commande (seulement depuis le mode interactif)

Le tableau ci-dessous rappelle la façon de couper du texte. Chaque commande basée sur "d" stocke ce qui a été coupé dans le presse papier de *vim*. Ces commandes s'exécutent uniquement en mode interactif.

Commande	Effet
x	Supprime le caractère sous le curseur
s	Supprime le caractère sous le curseur puis entre en mode insertion
u	Annule la dernière action
[n]dd ou d[n]d	Supprime n lignes à partir de celle où est placé le curseur. Si n n'est pas spécifié, supprime une ligne
[n]dw	Supprime n mots à partir de celui où est placé le curseur. Si n n'est pas spécifié, supprime un mot
d0	Supprime depuis le début de la ligne courante jusqu'au curseur
d\$	Supprime depuis le curseur jusqu'à la fin de la ligne courante
dgg	Supprime du début du document, jusqu'au curseur
dG	Supprime du curseur jusqu'à la fin du document

Commandes de copie :

Commande	Effet
[n]yy ou y[n]y ou [n]Y	Copie les n lignes à partir de celle du curseur (si n n'est pas spécifié, copie une seule ligne)
[n]yw ou y[n]w	Copie les n mots à partir de celui du curseur (si n n'est pas spécifié, copie un seul mot)
y0	Copie entre le début de la ligne et le curseur
y\$	Copie entre le curseur et la fin de la ligne
ygg	Copie depuis le début du document, jusqu'au curseur
yG	Copie depuis le curseur jusqu'à la fin du document
p	Colle après le du curseur
P	Colle avant le curseur

Sauvegarder et quitter :

Commande	Effet
:w [nom de fichier]	Enregistrer le fichier (si <i>nom de fichier</i> est spécifié, enregistre dans ce fichier)
:q	Quitte <i>vim</i> si aucune modification non enregistrée n'existe
:q!	Force à quitter <i>vim</i> en abandonnant les modifications non enregistrées
:wq ou :x	Enregistre et quitte

Divers :

Commande	Effet
:set number	Affiche les numéros de lignes
:vsp ou :vsplit	Sépare l'écran verticalement
:sp ou :split	Sépare l'écran horizontalement
<b>Ctrl-w</b> puis l	Va vers le "split" de droite
<b>Ctrl-w</b> puis h	Va vers le "split" de gauche
<b>Ctrl-w</b> puis j	Va vers le "split" du bas
<b>Ctrl-w</b> puis k	Va vers le "split" du haut