

Préparation LPI

Exam 101

103.8 Vi

- Poids : 2
- Se déplacer dans u document avec vi
- Savoir utiliser les modes de vi
- Insérer, modifier, supprimer, copier et trouver du texte

- Attention : en fonction des UNIX ou des différentes distributions
-> comportement différent de vi/vim
 - Certaines distributions ne conservent que vim et la commande vi n'est autre que vim
 - D'autres conservent une distinction entre les deux commandes (Ubuntu)

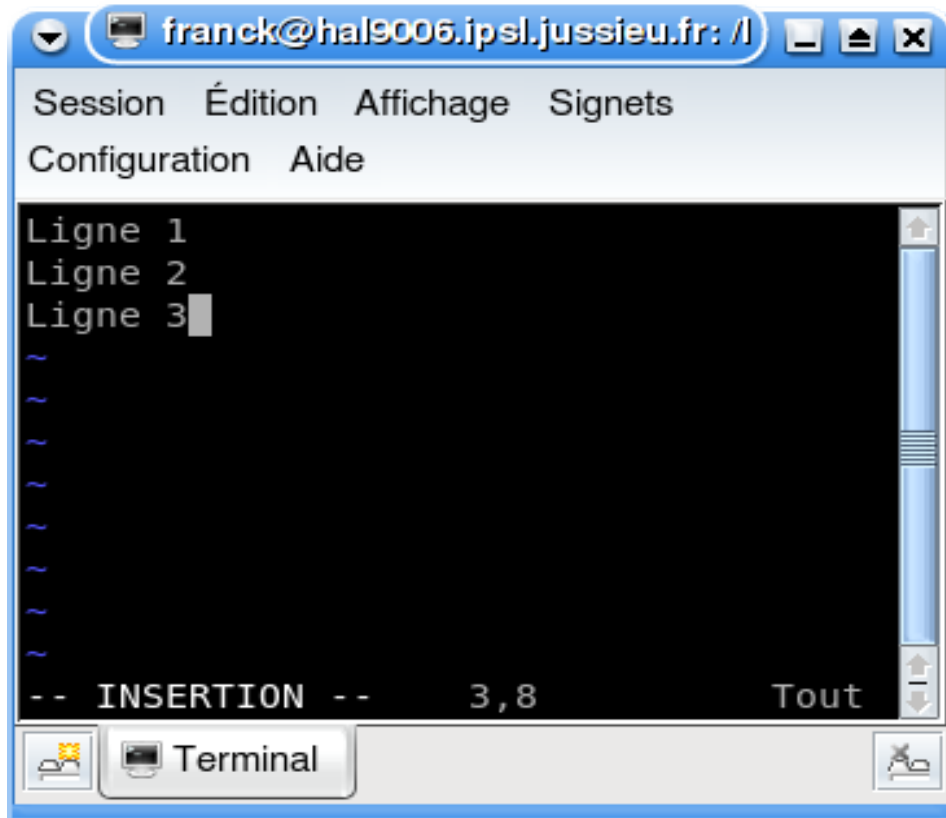
- Vi fonctionne sous **3 modes**
 - **Commande** : saisie possible d'une commande (souvent une touche) pour agir sur le document.
C'est le mode par défaut au démarrage
 - **Edition** (*Insert*): permet la saisie du texte
 - **Ex** (*mode dernière ligne*) : saisie des commandes relatives à la gestion de fichiers (entre autres)
- Vi est un éditeur en mode texte
 - vous utilisez le clavier à la fois pour éditer le texte du document (insérer-supprimer des caractères) et pour lancer les commandes du logiciel (gestion des fichiers par exemple)

- Le plus difficile au début : savoir dans quel mode on se trouve !
 - voir le mode en bas de l'écran à gauche.
INSERT ou INSERTION apparaîtra si vous êtes en mode Edition
 - s'il n'y a rien, vous êtes en mode Commande
 - si le curseur est en bas de l'écran après « : », vous êtes en mode Ex

- Démarrer vi

- `$ vi` : démarre vi sur un nouveau document

- `$ vi exemple.txt` : démarre vi en ouvrant le document *exemple.txt*



- Commandes passant en mode saisie de texte

Séquence	Action
i	Insérer à la position courante du curseur
a	Insérer à la position suivante du curseur
I	Insérer en début de ligne
A	Insérer en fin de ligne
o	Ouvrir une nouvelle ligne en dessous du curseur
O	Ouvrir une nouvelle ligne au dessus du curseur
cw	Changer un mot
c\$	Changer jusqu'à la fin de la ligne

- Sortie du mode saisie de texte et passage au mode Commande
 - touche ESC

- Ctrl + F : affiche l'écran suivant
- Ctrl + B : affiche l'écran précédent
- Ctrl + D : affiche de 1/2 écran suivant
- Ctrl + U : affiche de 1/2 écran précédant

- Commandes principales

Séquence	Action
x	détruire le caractère sous le curseur
r	remplacer le caractère sous le curseur
dd	effacer la ligne courante
dw	effacer le mot sous le curseur
.	répéter la dernière commande
J	joindre la ligne suivante avec la ligne courante
/ toto n	rechercher le mot toto dans le texte en avant répéter la dernière recherche
u	annulation de la dernière commande (undo)
CTRL + L	rafraîchir l'écran

- *nombre* : supprime nombre lignes à partir de la ligne courante
- Attention : la ligne courant compte
 - 5dd : supprime la ligne courante et les 4 lignes qui la suivent

- Commandes de sauvegarde et de sortie de vi

Séquence	Action
:w	sauver le fichier éditer
:w toto	sauver le fichier dans toto
:q	quitter vi
:q!	quitter vi sans sauvegarder
:wq ou ZZ	sauver puis quitter vi
:e toto	éditer maintenant le fichier toto
:r toto	importer maintenant le fichier toto
:e!	éditer la copie courante du fichier (abandon des m

- Attention : toutes ces commandes se font en mode Ex
 - touche ESC + « : » pour passer en mode Ex

- Commandes de copier/coller

Séquence	Action
yy	copie la ligne courante
p	colle dans le texte le contenu du presse-papier
nombre y	copie nombre lignes
:n,my	copie de la ligne n à la ligne m (mode Ex)

- Attention : la ligne courant compte

- 5yy : copie la ligne courante et les 4 lignes qui la suivent

- Commandes de substitution

Séquence	Action
<code>:s/toto/titi/</code>	sur la ligne du curseur, remplacer le premier «toto» par «titi»
<code>:s/toto/titi/g</code>	sur la ligne du curseur, remplacer tous les «toto» par «titi»
<code>:1,\$s/toto/titi</code>	de la ligne 1 à la dernière ligne (\$), remplacer mot «toto» par «titi»
<code>:1,\$s/toto/titi/g</code>	de la ligne 1 à la dernière ligne (\$), remplacer mots «toto» par «titi»
<code>:%s/toto/titi/g</code>	remplace de la première à la dernière ligne c «toto» par «titi»

- Principales options

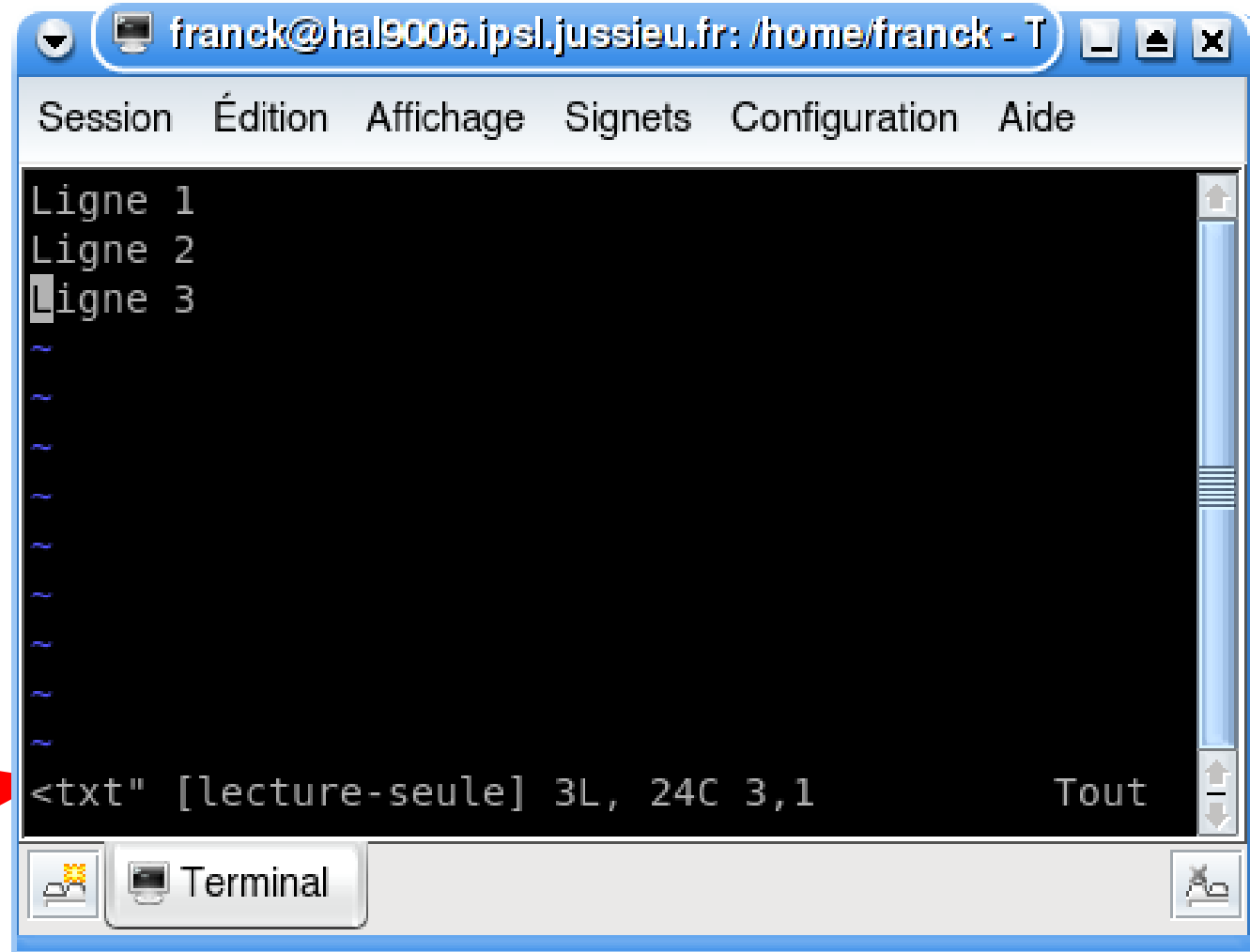
Séquence	Action
<code>:set all</code>	affiche toutes les options possibles
<code>:set opt</code>	positionne l'option <code>opt</code> à vrai
<code>:set noopt</code>	positionne l'option <code>opt</code> à faux
<code>:set nu</code>	affiche les numéros de ligne
<code>:set nonu</code>	enlève les numéros de ligne

- Divers

- en cas de plantage de `vi`, utiliser la commande `vi -r exemple.txt` pour essayer de récupérer le document
- pour consulter un fichier sans le modifier : `vi -R exemple.txt`

- Divers ...
 - Pour récupérer le contenu du fichier en cas de plantage :
création de fichier caché temporaire
 - vi toto
 - .toto.swp

- Pratique : `view` lance `vi` en read only



- Encore plus pratique ! : vim est livré avec un tutoriel
- Il suffit de lancer 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).

