



TP 4 Linux

Les éditeurs de texte sous Shell



Plan du cours

1. Définition d'un éditeur texte
2. Pourquoi a-t-on besoin des éditeurs de texte sous Linux
3. L'éditeur de texte : nano
4. L'éditeur de texte : Vim



Définition d'un éditeur de texte

- Un éditeur de texte est un programme qui permet de modifier des fichiers de texte brut, sans mise en forme (gras, italique, souligné...).
- Sous Windows, on dispose d'un éditeur de texte : le Bloc-Notes. Sous Linux, on a le choix entre Nano, Vim, Emacs et bien d'autres, sachant qu'au moins un de ceux-là est installé par défaut sur la plupart des distributions.



Définition d'un éditeur de texte

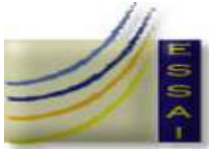
- Les éditeurs de texte sous linux sont parfaits pour les programmeurs en particulier : ils permettent d'éditer des fichiers .c, .cpp, .h, .rb, .py, etc.
- Même si vous ne programmez pas, vous aurez besoin d'utiliser un éditeur de texte pour modifier les fichiers de configuration du linux.



Présentation de l'éditeur nano

- Introduction

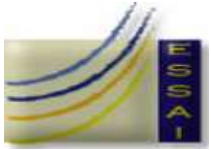
- En sciences, le terme « nano » représente une toute petite unité. Par exemple, un atome a une taille d'environ 0,1 nanomètre.
- L'éditeur nano est un programme très simple comparé à Vim et Emacs et il vous conviendra tout à fait pour démarrer.
- Le nom complet de Nano est « GNU nano », en référence au projet GNU . Il s'agit d'un logiciel qui s'inspire de « pico », un éditeur de texte plus ancien qui se voulait lui aussi très simple d'utilisation



L'éditeur nano

- Lancer l'éditeur nano
 - Pour lancer cet éditeur taper la commande **nano** puis entrée

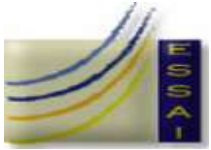
```
Terminal
administrateur@administrateur-VirtualBox: ~
administrateur@administrateur-VirtualBox:~$ nano
```



L'éditeur nano

- L'éditeur s'ouvre automatiquement

- Maintenant on peut commencer à taper notre texte



L'éditeur nano

- Les raccourcis clavier de nano
 - En bas de la fenêtre nano vous trouvez une liste des raccourcis qui peut vous aider à manipuler l'éditeur

<code>^G</code> Aide	<code>^O</code> Écrire	<code>^R</code> Lire fich.	<code>^Y</code> Page préc.	<code>^K</code> Couper	<code>^C</code> Pos. cur.
<code>^X</code> Quitter	<code>^J</code> Justifier	<code>^W</code> Chercher	<code>^V</code> Page suiv.	<code>^U</code> Coller	<code>^T</code> Orthograp.

- Le symbole ^ signifie Ctrl.

Exemple : pour quitter Nano, il suffit de taper Ctrl + X.



L'éditeur nano

- Voici les raccourcis les plus importants:
 - **Ctrl + K** : couper la ligne de texte
 - **Ctrl + U** : coller la ligne de texte que vous venez de couper
 - **Ctrl + C** : afficher à quel endroit du fichier votre curseur est positionné (numéro de ligne...)
 - **Ctrl + W** : rechercher dans le fichier
 - **Ctrl + O** : enregistrer le fichier (écrire)
 - **Ctrl + X** : quitter Nano



L'éditeur nano

- Vous pouvez vous déplacer dans le fichier avec les flèches du clavier ainsi qu'avec les touches Page Up et Page Down pour avancer de page en page
- Si l'aide-mémoire vous encombre, vous pouvez gagner de la place en appuyant sur Échap puis sur X. Vous pouvez l'afficher de nouveau avec la même suite de touches.



L'éditeur nano

- La recherche avec l'éditeur nano
- Si vous voulez chercher un mot dans un fichier ouvert avec l'éditeur nano:
 - Taper ctrl + w
 - Puis taper votre mot puis entrée
 - Si le mot existe sa première lettre va être sélectionnée

Voir les graphiques suivants:



L'éditeur nano

```
GNU nano 2.2.6          Nouvel espace          Modifié
Voici mon
premier texte avec un editeur Linux
```

1. taper ctrl + w pour lancer la recherche

2. une zone de recherche s'affiche

3. taper ton mot puis appuyer sur entrée

```
Recherche [r]: texte
^G Aide      ^Y Prem. lig.^T Aller lig.^W Début paraM-J JustifierM-B ->Arrière
^C Annuler   ^V Dern. Lig.^R Remplacer ^O Fin para. M-C Resp.cassM-R Exp. ratio.
```

- Remarque: la recherche précédente lancée avec l'éditeur nano, est mémorisée et mise entre crochet. dans notre cas est **[r]**



L'éditeur nano

```
GNU nano 2.2.6          Nouvel espace          Modifié
Voici mon
premier texte avec un editeur Linux
```

6. Changer la position du curseur, si vous ne voulez pas faire une nouvelle recherche

4. La première lettre de première occurrence du mot cherché sera sélectionnée

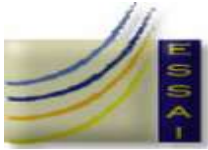
5. L'éditeur il vous donne la main pour lancer une nouvelle recherche

```
[ La recherche a fait le tour ]
^G Aide      ^O Écrire    ^R Lire fich.^Y Page préc.^K Couper      ^C Pos. cur.
```



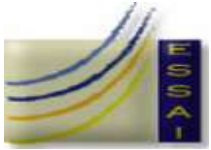
L'éditeur nano

- Enregistrer et sortir de l'éditeur
 - Si vous voulez enregistrer un texte édité par nano,
taper `ctrl + o`
 - Suivre les graphiques suivants pour comprendre
comment enregistrer et quitter nano



L'éditeur nano

```
administrateur@administrateur-VirtualBox: ~  
GNU nano 2.2.6          Nouvel espace          Modifié  
Voici mon premier texte avec une editeur nano  
  
1. Taper Ctrl + o pour enregistrer  
un fichier  
  
2. Taper le nom du fichier puis  
appuyer sur entrée  
  
Le nom du fichier  
  
Nom du fichier à écrire: t2  
^G Aide          M-D Format DOS    M-A Ajout (à la fin) M-B Copie de sécu.  
^C Annuler       M-M Format Mac    M-P Ajout (au début)
```



L'éditeur nano

```
administrateur@administrateur-VirtualBox: ~
GNU nano 2.2.6      Fichier : t2      Modifié
Voici mon premier texte avec une editeur nano
nano est editeur simple

1. Taper Ctrl + x pour sortir du
fichier

2. Si vous taper o vous passer au mode
enregistrer sinon n pour sortir

Sauver l'espace modifié (RÉPONRE « Non » EFFACERA LES CHANGEMENTS) ?
O Oui
N Non
^C Annuler
```




L'éditeur nano

- Les paramètres de la commande nano:
 - Lorsque vous appelez Nano dans la ligne de commandes, vous pouvez spécifier plusieurs paramètres. Le plus courant est d'indiquer en paramètre le nom du fichier qu'on veut ouvrir.

Exemple:

```
administrateur@administrateur-VirtualBox: ~  
administrateur@administrateur-VirtualBox:~$ nano t2
```

- Si t2 n'existe pas il va être créé, sinon il va être ouvert pour l'édition



L'éditeur nano

- La commande nano peut avoir plusieurs options, on peut citer les plus utiles:
 - -m : autorise l'utilisation de la souris sous Nano.
 - -i : indentation automatique. Très utile lorsque vous éditez un fichier de code source.
 - -A : active le retour intelligent au début de la ligne. Normalement, lorsque vous appuyez sur la touche Home située à côté de la touche Fin, le curseur se repositionne au tout début de la ligne. Avec cette commande, il se positionnera après les alinéas.



L'éditeur nano

- Exemple de création du fichier de configuration avec l'éditeur nano
 - Sous linux, les fichiers de configuration commencent par `.` et ils sont cachés dans le répertoire où ils existent, pour les afficher taper la commande **ls -a**
 - Dans notre cas, nous allons créer un fichier `.nanorc` sous notre répertoire personnel qui sert à changer quelques options de l'éditeur nano
 - Pour se faire, suivre les graphiques suivants:



L'éditeur nano

administrateur@administrateur-VirtualBox: ~
administrateur@administrateur-VirtualBox:~\$ nano .nanorc

1. Commencez par créer le fichier

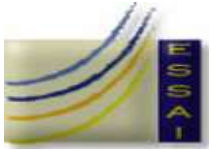
GNU nano 2.2.6 Fichier : .nanorc Modifié

set mouse
set autoindent
set smarthome

4. L'éditeur nano va prendre par défaut les options: souris, indentation automatique et retour intelligent au debut de ligne

2. Taper les instructions suivantes

3. Pour ajouter des options par défaut à nano, utiliser la commande **set**

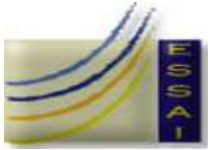


L'éditeur nano

- Enregistrer le fichier et sortir de l'éditeur en tapant `ctrl+x`
- Taper une autre fois la commande `nano` pour que l'éditeur s'ouvre avec ses nouvelles options
- Pour voir les différentes options qu'on peut les appliquer par défaut sur notre éditeur nano, utiliser la commande suivante:

`sudo nano /etc/nanorc`

Voir le graphique suivant:



L'éditeur nano

```
GNU nano 2.2.6      Fichier : /etc/nanorc
# punctuation, optionally followed by closing brackets,
## sentences.
##
# set brackets "'">]}"

## Do case sensitive searches by default.
# set casesensitive

## Constantly display the cursor position in the statusbar. Note that
## this overrides "quickblank".
# set const

## Use cut to end of line by default.
# set cut
```

1. dans cet fichier,
vous trouvez toutes
les options

2. Si vous voulez ajouter une option
à votre .nanorc, copier l'instruction
sans #

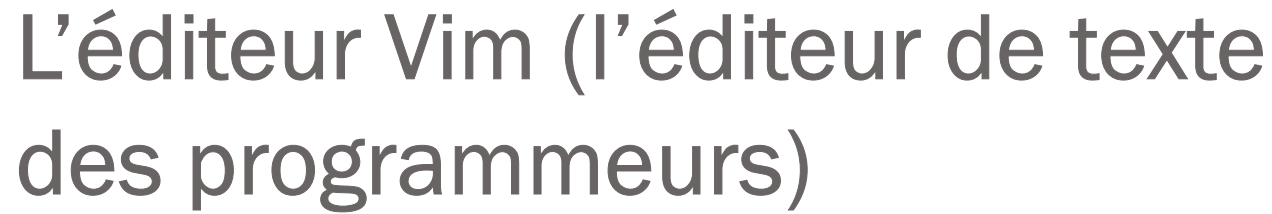
3. une instruction qui commence
par # est un commentaire



L'éditeur Vim (l'éditeur de texte des programmeurs)



- C'est quoi Vim
 - Vim : il s'agit d'une version améliorée de l'un des plus anciens éditeurs en console : « Vi »
 - Vim (VI iMproved , version améliorée de Vi) est largement répandu et généralement disponible par défaut sur la plupart des OS basés sur Unix, comme Linux.



- Pour lancer l'éditeur Vim, taper la commande **vi** ou **vi** **[nom_fichier]**

- ```
adminstrateur@adminstrateur-VirtualBox: ~
~
~
VIM - Vi IMproved
~
version 7.3.429
by Bram Moolenaar et al.
Modified by pkg-vim-maintainers@lists.alioth.debian.org
Vim is open source and freely distributable

Become a registered Vim user!
type :help register<Enter> for information

type :q<Enter> to exit
type :help<Enter> or <F1> for on-line help
type :help version7<Enter> for version info

Running in Vi compatible mode
type :set nocompatible<Enter> for Vim defaults
type :help cp-default<Enter> for info on this
```
- recording





# L'éditeur Vim (l'éditeur de texte des programmeurs)



- Vim possède 3 modes différents:
  - Mode interactif : c'est le mode par défaut par lequel vous commencez. En lançant Vim. Dans ce mode, vous ne pouvez pas écrire de texte . Le mode interactif est un mode puissant qui permet de se déplacer dans le texte, de supprimer une ligne, copier-coller du texte, rejoindre une ligne précise, annuler ses actions, etc.
  - Chaque action peut être déclenchée en appuyant sur une touche du clavier (par exemple, on appuie sur u pour annuler la dernière action).



# L'éditeur Vim (l'éditeur de texte des programmeurs)



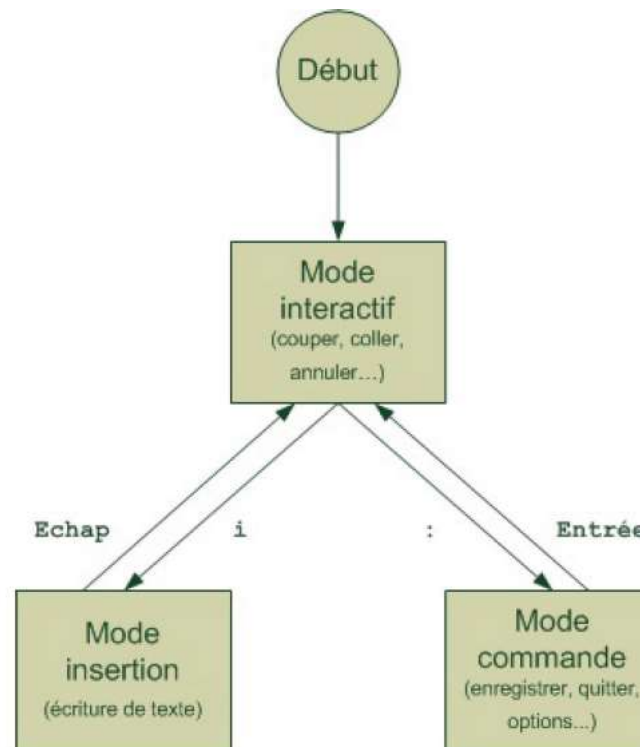
- Mode insertion : Vous tapez du texte et ce dernier s'insère à l'endroit où se trouve le curseur. Pour entrer dans ce mode, il existe plusieurs possibilités. L'une des plus courantes est d'appuyer sur la touche i. Pour en sortir, il faut appuyer sur la touche Echap.
- Mode commande : ce mode permet de lancer des commandes telles que « quitter », « enregistrer », etc. Vous pouvez aussi l'utiliser pour activer des options de Vim (comme la coloration syntaxique, l'affichage du numéro des lignes...).



# L'éditeur Vim (l'éditeur de texte des programmeurs)



- Voici un graphique qui résume les différents modes de l'éditeur Vim





# L'éditeur Vim (l'éditeur de texte des programmeurs)



- Les commandes sous Vim
  - Avant d'utiliser ses commandes, taper Echap:
  - **Les commandes de saisie**

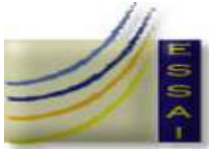
| Commande | Action                                                      |
|----------|-------------------------------------------------------------|
| a        | Ajout de texte derrière le caractère actif                  |
| A        | Ajout de texte en fin de ligne                              |
| i        | Insertion de texte devant le caractère actif                |
| I        | Insertion de texte en début de ligne                        |
| o        | Insertion d'une nouvelle ligne sous la ligne active         |
| O        | Insertion d'une nouvelle ligne au-dessus de la ligne active |



# L'éditeur Vim (l'éditeur de texte des programmeurs)



- **Les commandes qui permettent de quitter l'éditeur:**
  - La commande ZZ quitte et sauve le fichier
  - :q! quitte sans sauver
  - :q quitte si le fichier n'a pas été modifié
  - :w sauve le fichier
  - :wq ou x sauve et quitte

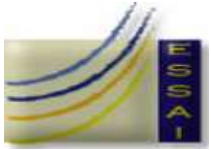


# L'éditeur Vim (l'éditeur de texte des programmeurs)



- Les commandes de déplacement dans le fichiers

| Commande  | Action                                     |
|-----------|--------------------------------------------|
| h         | Vers la gauche                             |
| i         | Vers la droite                             |
| k         | Vers le haut                               |
| j         | Vers le bas                                |
| 0 (zéro)  | Début de ligne (:0 première ligne)         |
| \$        | Fin de ligne (:\$ dernière ligne)          |
| w         | Mot suivant                                |
| b         | Mot précédent                              |
| <u>fc</u> | Saut sur le caractère 'c'                  |
| Ctrl + F  | Remonte d'un écran                         |
| Ctrl + B  | Descend d'un écran                         |
| G         | Dernière ligne du fichier                  |
| NG        | Saute à la ligne 'n' ( <u>n</u> identique) |



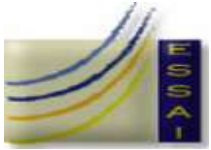
# L'éditeur Vim (l'éditeur de texte des programmeurs)



- **Les commandes de correction**

| Commande   | Action                                                                          |
|------------|---------------------------------------------------------------------------------|
| x          | Efface le caractère sous le curseur                                             |
| X          | Efface le caractère devant le curseur                                           |
| <u>rc</u>  | Remplace le caractère sous le curseur par le caractère 'c'                      |
| <u>dw</u>  | Efface le mot depuis le curseur jusqu'à la fin du mot                           |
| d\$ (ou D) | Efface tous les caractères jusqu'à la fin de la ligne                           |
| <u>dO</u>  | Efface tous les caractères jusqu'au début de la ligne.                          |
| <u>dfc</u> | Efface tous les caractères de la ligne jusqu'au caractère 'c'                   |
| dG         | Efface tous les caractères jusqu'à la dernière ligne, ainsi que la ligne active |
| <u>D1G</u> | Efface tous les caractères jusqu'à la première ligne, ainsi que la ligne active |
| dd         | Efface la ligne active                                                          |

Pour annuler une action, taper u



# L'éditeur Vim (l'éditeur de texte des programmeurs)



- **Les commandes de recherche:**
  - Contrairement à un éditeur de texte classique, vi peut rechercher autre chose que des mots simples et fonctionne à l'aide de caractères spéciaux et de critères. La commande de recherche est le caractère « / ».
  - La recherche démarre du caractère courant à la fin du fichier. Le caractère « ? » effectue la recherche en sens inverse. On indique ensuite le critère, puis Entrée.





# L'éditeur Vim (l'éditeur de texte des programmeurs)



Exemple : /echo

- recherche la chaîne 'echo' dans la suite du fichier. Quand la chaîne est trouvée, le curseur s'arrête sur le premier caractère de cette chaîne.
- La commande « n » permet de continuer la recherche dans le sens indiqué au début.
- La commande « N » effectue la recherche en sens inverse.



# L'éditeur Vim (l'éditeur de texte des programmeurs)



- **Quelques critères de recherche**
  - `/[FfBb]oule` : Foule, foule, Boule, boule
  - `/[A-Z]e` : Tout ce qui commence par une majuscule avec un e en deuxième position.
  - `/[A-Za-Z0-9]` : tout ce qui commence par une majuscule, minuscule ou un chiffre
  - `/[^a-z]` : plage négative : tout ce qui ne commence pas par une minuscule
  - `/vé.o` : le point remplace un caractère, vélo, véto, véro, ...



# L'éditeur Vim (l'éditeur de texte des programmeurs)



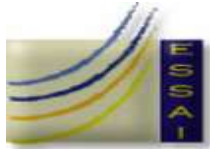
- /Au\*o : l'étoile est un caractère de répétition, de 0 à n caractères, Auo, Auto, Automoto, ...
- /.\* : l'étoile devant le point, une chaîne quelconque de taille variable
- /^Auto : le ^ indique que la chaîne recherchée devra être en début de ligne
- /Auto\$ : le \$ indique que la chaîne recherchée devra être en fin de ligne



# L'éditeur Vim (l'éditeur de texte des programmeurs)



- `/Au*o` : l'étoile est un caractère de répétition, de 0 à n caractères, `Auo`, `Auto`, `Automoto`, ...
- `/.*` : l'étoile devant le point, une chaîne quelconque de taille variable
- `/^Auto` : le `^` indique que la chaîne recherchée devra être en début de ligne
- `/Auto$` : le `$` indique que la chaîne recherchée devra être en fin de ligne



# L'éditeur Vim (l'éditeur de texte des programmeurs)



- Quelques commandes de remplacement

| Commande                     | Action                                                              |
|------------------------------|---------------------------------------------------------------------|
| <code>cw</code>              | Remplacement du mot courant                                         |
| <code>c\$</code>             | Remplacement jusqu'à la fin de la ligne                             |
| <code>cO</code>              | Remplacement jusqu'au début de la ligne                             |
| <code>cfx</code>             | Remplacement jusqu'au prochain caractère 'x' dans la ligne courante |
| <code>c/Auto (Entrée)</code> | Remplacement jusqu'à la prochaine occurrence de la chaîne 'Auto'    |

- Après cette saisie, le caractère \$ apparaît en fin de zone à modifier.

Il suffit alors de taper son texte et d'appuyer sur Echap.



# L'éditeur Vim (l'éditeur de texte des programmeurs)



- **Les commandes de copier/ coller**
  - On utilise la commande « y » (Yank) pour copier du texte, elle-même devant être combinée avec d'autres indications.
  - Pour couper (déplacer), c'est la commande « d ».
  - Pour coller le texte à l'endroit choisi, c'est la commande « p » (derrière le caractère) ou « P » (devant le caractère).  
Si c'est une ligne complète qui a été copiée, elle sera placée en-dessous de la ligne active



# L'éditeur Vim (l'éditeur de texte des programmeurs)



- **Les commandes de copier/ coller**

- On utilise la commande « y » (Yank) pour copier du texte, elle-même devant être combinée avec d'autres indications.
- Pour couper (déplacer), c'est la commande « d ».
- Pour coller le texte à l'endroit choisi, c'est la commande « p » (derrière le caractère) ou « P » (devant le caractère).  
Si c'est une ligne complète qui a été copiée, elle sera placée en-dessous de la ligne active



# L'éditeur Vim (l'éditeur de texte des programmeurs)



- **Fusion des fichiers**
  - On peut insérer un fichier dans une position bien déterminé du curseur, pour se faire, taper la commande : **:r nom\_fichier**
- **Lancer une commande externe**
  - Il est possible d'écrire des commandes traditionnelles du shell directement dans Vim. Pour cela, commencez par taper **:!** suivi du nom de la commande. Essayez par exemple de taper **:!ls**. Vous afficherez alors le contenu du dossier dans lequel vous vous trouvez Cette fonctionnalité est bien pratique pour effectuer quelques actions sans avoir à quitter Vim.





# Merci pour votre attention

Contact: [sihemmansour@yahoo.fr](mailto:sihemmansour@yahoo.fr)