

Introduction à l'édition de texte avancée : l'exemple de GNU Nano

Par mist



www.openclassrooms.com

Sommaire

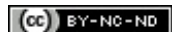
Sommaire	2
Introduction à l'édition de texte avancée : l'exemple de GNU Nano	3
Un peu de théorie	3
Se déplacer avec Nano	4
Opérations sur le texte	5
La marque et la région	5
Quelques commandes supplémentaires	6
Q.C.M.	6
Partager	7

η Introduction à l'édition de texte avancée : l'exemple de GNU Nano

Par mist

Mise à jour : 04/09/2009

Difficulté : Intermédiaire



Bonjour à tous les zér0s !

Ce tutoriel est une sorte de suite au tuto de M@teo21 sur l'éditeur de texte GNU Nano. Vous devez avoir lu celui-ci pour suivre mon tutoriel.

Vous apprendrez ici à éditer très rapidement vos fichiers grâce à Nano, en utilisant les commandes les plus puissantes !

Nano n'est pour moi qu'un prétexte pour introduire l'édition de texte avancée. J'ai choisi Nano car en prenant des éditeurs comme Vim ou Emacs, je perdrais trop de temps à expliquer les commandes au lieu d'expliquer la manière de les utiliser ! Beaucoup de tutoriels sur le Web ne présentent qu'une liste de commandes sans trop d'explications sur la stratégie d'édition. Nano me permet d'expliquer ces bases sans rentrer dans des détails techniques inutiles.

C'est pourquoi même si Nano ne vous dit rien, ce tutoriel peut quand même vous intéresser ! 😊

Sommaire du tutoriel :



- [Un peu de théorie](#)
- [Se déplacer avec Nano](#)
- [Opérations sur le texte](#)
- [Q.C.M.](#)

Un peu de théorie

Nous allons d'abord répondre à quelques questions courantes avant de commencer.



Pourquoi utiliser un éditeur de texte en console ? Mon éditeur en interface graphique, avec la souris, me convient très bien.

Si vous dites ça, c'est parce que vous ne vous êtes jamais vraiment investi dans l'apprentissage d'un éditeur de texte digne de ce nom. A partir du moment où on essaye, on devient rapidement accro ! 😊

La raison est que des éditeurs puissants en console permettent d'être bien plus rapide. Vous ne déplacez jamais vos mains (ni pour aller jusqu'à la souris, ni pour aller jusqu'aux flèches directionnelles), et des opérations d'édition peuvent être exécutées à une vitesse impressionnante pour le néophyte.



Pourquoi utiliser Nano ? Emacs et Vim sont beaucoup plus puissants !

Il m'est assez difficile de répondre à cette question, étant donné qu'objectivement Vim et Emacs sont effectivement plus puissants.

Personnellement, je suis un fana de Vim, et pourtant comme vous le voyez j'écris un tuto sur Nano ! Il y a plusieurs raisons, mais parmi elles il y a le fait que Nano est un bon compromis entre la simplicité d'utilisation et l'efficacité. Pour les débutants, il est à mi-chemin entre un éditeur graphique (comme gedit) et un éditeur complexe (comme Vim/Emacs).

Par ailleurs, je considère que pour faire un choix d'éditeur, il faut en essayer plusieurs, sinon ce n'est pas réellement un choix !



J'ai déjà entendu parlé des modes. Qu'est-ce que c'est ?

Globalement, il y a deux types d'éditeurs de texte :

- Ceux basés sur les modes (ex : Vim). Dans ce genre d'éditeur, vous alternez entre le mode d'insertion et le mode de commande. L'intérêt est qu'en mode commande des touches simples du clavier peuvent être utilisées (**d** pour couper, **y** pour coller, etc)
 - Ceux qui sont complètement interactifs (ex : Emacs). Ce genre d'éditeur ressemble beaucoup aux éditeurs graphiques, car toutes les lettres entrées au clavier sont effectivement insérées dans le texte. Vous tapez du texte, il s'affiche du texte ! 😊
- Pour utiliser des commandes, vous devez rester appuyer sur **Ctrl** ou **Alt**.

Les deux types d'éditeur ont leurs détracteurs, et il n'y en a pas l'un meilleur que l'autre : cela dépend de vos goûts. Comme je l'ai déjà dit, il vaut mieux essayer les deux avant de choisir !

Quant à Nano, c'est un éditeur complètement interactif : le concept de mode n'existe pas sous Nano.



D'après ce que tu m'a dis avant, j'en déduis qu'il va falloir apprendre beaucoup de commandes. Cela va me faire perdre du temps et de l'énergie !

Pour utiliser basiquement un éditeur de texte, une soirée d'apprentissage suffit. Ensuite, il faut compter environ une à deux semaines pour maîtriser vraiment correctement Vim ou Emacs : deux ou trois jours suffisent pour Nano.

Notez que dans tous les cas, cela vaut le coup : perdre du temps pour en regagner cent fois plus après !

Maintenant toutes ces questions réglées, nous pouvons apprendre à éditer du texte *pour de vrai*. 😊

Se déplacer avec Nano

Nous allons commencer par passer en revue les commandes de déplacement sous Nano.

Je note les commandes ainsi : **^X** signifie rester appuyer sur CTRL puis appuyer sur X. **M-X** signifie rester appuyer sur Meta et appuyer sur X. La touche Meta dépend de votre système, c'est en général soit Alt soit Echap.

Les quatre premières commandes à connaître sont :

- **^P** déplace le curseur vers le haut
- **^N** déplace le curseur vers le bas
- **^F** déplace le curseur vers la droite
- **^B** déplace le curseur vers la gauche



Ça sert à quoi ? J'ai déjà les flèches directionnelles !

Là il va falloir me croire sur parole : passé un temps d'adaptation vous serez bien plus rapide avec les commandes ci-dessus. En effet, les touches directionnelles sont éloignées à droite du clavier, et il faut déplacer sa main pour les atteindre à chaque fois ! En bon utilisateur de Nano, je vous conseille de ne jamais utiliser les flèches.

Mais heureusement, vous n'aurez pas à utiliser souvent ces touches, car il existe d'autres commandes de déplacement très pratiques. Notamment :

- **^Space** : se déplace d'un mot en avant
- **M-Space** : se déplace d'un mot en arrière
- **^Y** : écran précédent
- **^V** : écran suivant

Il faut savoir que les *power users* (c'est comme ça que l'on appelle les utilisateurs les plus rapides) ne se servent presque jamais

des commandes de déplacement caractères par caractères. Il est plus rapide de se déplacer directement à l'endroit voulu. C'est pour cela qu'on a à notre disposition des puissantes commandes de déplacement !

Ces commandes de saut sont aussi très importantes :

- **^A** saute au début de la ligne
- **^E** saute à la fin de la ligne
- **M-(** saute au début du paragraphe
- **M-)** saute à la fin du paragraphe
- **M-** saute au début du fichier
- **M-/** saute à la fin du fichier

Mais ces commandes ne suffisent pas toujours : lorsqu'on veut se déplacer à un endroit précis du texte par exemple. En ce cas, nous allons utiliser une fonction chère aux programmeurs, la fonction de recherche.

Pour rechercher une chaîne de caractère, il suffit d'utiliser **^W** puis de taper la chaîne et enfin **Enter**. Nano vous déplacera jusqu'à la première occurrence de la chaîne trouvée. Et pour répéter la dernière recherche effectuée, il y a la commande **M-W**.

Attention ! Par défaut, les chaînes de caractères à rechercher sont des expressions rationnelles. Si vous ne savez pas ce que c'est ou si vous considérez que c'est inutile (je vous comprend), je vous conseille de désactiver les expressions rationnelles en ajoutant à votre fichier de configuration (`~/.nanorc`) la ligne suivante :

Code : Bash

```
unset regexp
```

Maintenant vous savez vous déplacer correctement dans Nano. Passons aux commandes de modification. 🧑

Opérations sur le texte

Vous connaissez le fameux CTRL-C CTRL-V ? 😊

Sous Nano, les commandes ne sont pas les mêmes, mais il y a bien moyen de faire des modifications basiques du texte comme couper, coller, etc.

- **^K** : Couper la ligne courante
- **M-^** : Copier la ligne courante
- **^U** : Coller



Quoi ? On ne peut faire que ça ?

Moi je veux couper un mot, une moitié de ligne, un paragraphe !

Arf, mais il faudrait pour cela un très grand nombre de commandes ! 😞 Heureusement, pour pallier à ce problème on dispose sous Nano de la **marque** et de la **région**.



Sous Emacs, en plus de disposer de la marque et de la région également, on a un grand nombre de commandes (ex : **M-D** qui supprime un mot). Sous Vim, les commandes de suppressions sont tellement complètes qu'on a même pas besoin de la marque et la région !

La marque et la région

La marque est une sorte de repère. On la place quelque part puis on se déplace. Pour poser la marque, on utilisera la commande **M-A**.

Essayez, pour voir ! Une fois la marque posée, lorsque vous vous déplacez l'espace entre la marque et le curseur est surligné. Cet espace est nommé la région.

Maintenant une révélation : si région il y a, **^K** et **M-^** n'agissent pas sur la ligne courante mais sur la région ! Grâce à ça, vous pouvez en quelque sorte "émuler" une sélection.

Observez la ligne suivante :

Code : Autre

```
{c}har editor[] = "Emacs style";
```

La place du curseur est notée entre accolades. Le but est de remplacer le mot *Emacs style* par *Nano*. Voilà la solution la plus rapide (je crois) :

^E M-A M-Space M-Space ^K Nano;

Explicitement :

1. On saute en fin de ligne.
2. On pose la marque.
3. On revient deux mots en arrière.
4. On supprime la région.
5. On entre le texte de remplacement.

Comme vous le voyez, le principe de la marque et de la région permet de sélectionner ce qu'on veut.

Quelques commandes supplémentaires

Il y a aussi quelque chose de très utile et qui va vous servir souvent : le remplacement. Lorsque vous tapez **M-R**, on vous demande d'abord une chaîne à rechercher. Ensuite, on vous demande la chaîne de remplacement ! Pour chaque occurrence de la chaîne recherchée trouvée, on vous demandera si vous voulez la remplacer. Si vous répondez **y**, elle sera remplacée par la deuxième chaîne que vous avez entrée. 🧙

Voici encore quelques commandes de suppression utiles :

- **^D** : Coupe le caractère qui se trouve sous le curseur
- **M-T** : Coupe tout le texte de la position du curseur jusqu'à la fin du fichier

Je crois que nous en avons fini avec les commandes de modifications.

Q.C.M.

Le premier QCM de ce cours vous est offert en libre accès.
Pour accéder aux suivants

[Connectez-vous](#) [Inscrivez-vous](#)

Quel est la commande pour effectuer une recherche dans le texte, et se déplacer à la première occurrence trouvée ?

- ☐ ^R
- ☐ M-R
- ☐ ^W

Quel est la commande qui permet de poser la marque ?

- ☐ M-A
- ☐ ^A
- ☐ ^Space

Voici une situation (où les crochets représentent la place du curseur) :

Code : Autre

```
{B}onjour à tous les utilisateurs de Nano ! Comment allez-vous ?
```

Le but est d'arriver à ce texte-là :

Code : Autre

```
Bonjour à tous les zér0s ! Comment allez-vous ?
```

Quel est l'enchaînement de commande nécessaire ?

- ☐ M-Space M-Space M-Space M-Space ^K M-Space M-Space M-Space ^D zér0s !
- ☐ ^Space ^Space ^Space M-A ^Space ^Space ^Space ^K zér0s !
- ☐ ^E M-Space M-Space M-Space M-Space M-Space ^K zér0s !

Quelle commande copie la ligne courante (ou la région courante) sans la supprimer ?

- ☐ M-^
- ☐ ^X
- ☐ ^O

Correction !

[Statistiques de réponses au QCM](#)

Voilà, j'espère que ce tutoriel vous aura permis d'y voir plus clair dans l'utilisation de Nano et dans l'édition de texte en général.



Je prépare actuellement un tutoriel similaire, mais sur l'utilisation de l'éditeur Joe cette fois !

Partager

