

TP2 - Emacs, un éditeur de texte.

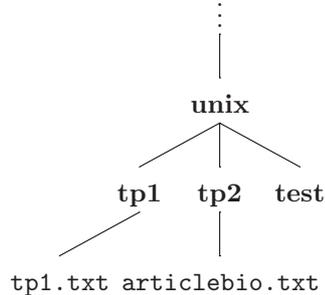
Révision Unix

Exercice 1 (Fichiers cachés).

1. Si vous avez une adresse que vous consultez régulièrement, vous pouvez vous transférer les mails qui arrivent à cette adresse. Pour cela, ouvrez `emacs` et créez un fichier `.forward` dans votre répertoire maison. Ce fichier doit simplement contenir l'adresse vers laquelle vos mails doivent être transférés.
2. Sur <http://www.pps.jussieu.fr/~tasson/enseignement/bioinfo/> vous trouverez les fichiers `.bashrc` et `.bash_profile` qui configurent votre terminal. Copiez-les dans votre répertoire maison.

Exercice 2 (Arborescence).

1. Placez-vous dans le répertoire `unix`.
2. Le fichier `articlebio.txt` se trouve dans le répertoire `/ens/tassonc`. Trouvez-le en explorant chacun des sous-répertoires ou à l'aide de la commande `find rep -name "nom"`
3. Créez l'arborescence suivante :



4. Placez-vous dans le répertoire `tp2` et lancez `emacs` tout en gardant la main dans le terminal.

Exercice 3 (Interrompre un programme, le lancer sans perdre la main).

1. Ouvrez le fichier texte `articlebio.txt` en utilisant la commande `emacs fichier`.
2. Une fois l'éditeur ouvert, revenez sur le terminal et tapez une commande. Que se passe-t-il ?
3. Toujours sous le terminal gardez la touche `[ctrl]` enfoncée puis pressez la touche `[c]`. Que se passe-t-il ?
4. Répétez les étapes 1 et 2 en ajoutant cette fois le caractère `&` en fin de commande (par exemple `emacs fichier &`). Voyez-vous une différence ?

Introduction à Emacs

Notations : Dans la suite du TP on utilisera toujours

- [C-touche] pour “en appuyant sur `ctrl` tapez sur la touche `touche`”.
- [M-touche] pour “en appuyant sur la touche `Esc` (ou `Esc`) tapez sur la touche `touche`”.
- [C-g] permet de quitter la ligne de commande en bas de votre fenêtre.

Emacs est un éditeur de texte très utilisé sous Unix. Toutes les opérations (sauvegarder, copier, coller...) peuvent être effectuées de deux manières différentes, en utilisant votre souris et le menus, ou en tapant des commandes directement au clavier. Vous verrez avec l’usage que vous gagnerez du temps en maîtrisant les raccourcis claviers.

Exercice 4 (Mais où est passée la souris?).

1. A l’aide du shell, ouvrez emacs. N’oubliez pas l’esperluette à la fin pour pouvoir continuer à taper des commandes dans le shell. Une fenêtre s’ouvre sur un fichier `*scratch*`, dans lequel aucune sauvegarde ne peut-être effectuée. Il est donc impératif de commencer par créer un nouveau fichier `txt1.txt` en cliquant sur `Save As`. Tout en bas de la fenêtre, apparaît une invite vous demandant le nom du fichier à sauvegarder.
2. Tapez une dizaine de lignes. Remarquez les `**` sur la barre en bas de la fenêtre. Sauvegardez en utilisant `Save Buffer` dans le menu `file`. A votre avis, à quoi servent les deux étoiles ?
3. Quelle est la différence entre `Save Buffer` et `Save Buffer As` ?
4. Tapez quelques mots supplémentaires puis quittez emacs (en passant par le menu `file`), que se passe-t-il ?

Nous allons maintenant recommencer cet exercice en utilisant les raccourcis clavier.

Exercice 5 (Emacs sans bouger les mains de son clavier (ou presque)).

1. Avec la souris, ouvrez le menu `file`, observez et notez les raccourcis clavier à droite des différentes commandes utilisées à l’exercice précédent. A présent, ne décollez plus les mains de votre clavier.
2. Tapez [C-x C-f]. En bas de votre fenêtre, une barre d’information vous demande d’entrer le nom du fichier que vous voulez créer, ici `txt2.txt`.
3. Tapez une dizaine de lignes et sauvegardez votre fichier (en utilisant un raccourci clavier).
4. Tapez à nouveau deux lignes et sauvegardez votre fichier sous le nom `txt2bis.txt` (toujours en utilisant un raccourci clavier).
5. Quittez emacs.

Un Buffer est un espace de stockage provisoire de données. Dans Emacs, on appelle donc Buffer tout endroit dans lequel l'utilisateur peut taper du texte qui n'est potentiellement pas sauvegardé (emacs signale d'une étoile les fichiers non sauvegardés).

Exercice 6 (Buffers, mini-buffers et tutti quanti).

1. Depuis un terminal (et à partir du dossier `tp2`), lancez `emacs` (n'oubliez pas `&` pour garder la main).
2. Chargez les fichiers `txt1.txt` et `txt2.txt`.
3. Dans le menu `buffer`, vous avez la liste des buffers ouverts. Quelle commande faut-il faire pour changer de buffer ? Passez dans le buffer `txt1.txt`.

En dessous du buffer, il existe deux barres :

- la barre d'information qui contient les informations sur l'état du buffer.
- le mini-buffer (en dessous de la barre d'information) dans lequel vous tapez les commandes (par exemple `[C-x C-s]` pour sauvegarder...) et dans lequel emacs interagit avec vous.

Exercice 7.

Tout en gardant un oeil sur la barre d'information, appuyez sur la touche `[Inser]`. Que se passe-t-il ? Modifiez le texte que vous avez tapé. Que remarquez-vous sur le mode d'édition ? Appuyez à nouveau sur cette touche pour revenir en mode normal.

Lorsque vous vous trompez de commande, vous pouvez toujours appuyer sur les touches `[C-g]` pour quitter le mini-buffer.

Multi-fenêtrage

Lorsque vous avez besoin de voir plusieurs fichiers à la fois, au lieu de lancer plusieurs fois emacs (ce qui a tendance à écrouler le système) vous pouvez diviser votre fenêtre (`Split Window`) en plusieurs sous-fenêtres ou ouvrir une nouvelle fenêtre (`New Frame`).

Exercice 8.

1. Allez voir dans le menu `file` les commandes décrites ci-dessus et essayez les. Notez les commandes clavier correspondantes.
2. *Attention la suite de l'exercice se fait sans décoller vos mains du clavier.* Ouvrez les fichiers `txt1.txt` et `txt2.txt` dans une même fenêtre coupée en deux. Utilisez la commande `[C-x o]` pour changer de mini-fenêtre. Puis revenez à la configuration initiale.
3. Ouvrez les mêmes fichiers mais cette fois dans deux fenêtres différentes. Utilisez la touche `[alt-tab]` pour changer de fenêtre. Revenez à la configuration initiale.

Commandes d'édition

Exercice 9.

1. Créez un nouveau fichier `mem_emacs.txt` dans le répertoire `tp2`.
2. Ouvrez le fichier `/ens/tassonc/texte1.txt` en divisant la fenêtre ou dans une nouvelle fenêtre (comme vous préférez!). *Attention* : Vous pouvez simplement lire ce fichier et non pas le sauver. Vous devez donc copier son contenu.
3. Recopiez, sans utiliser de commande, le contenu du fichier `texte1.txt` dans `mem_emacs.txt`, fermez la fenêtre du fichier `/ens/tassonc/texte1.txt`.
4. Recommencez les étapes 2 et 3 avec les fichiers `/ens/tassonc/textei.txt`, *i* variant de 2 à 4.
5. Remplir les trous en utilisant les commandes décrites.

Exercice 10.

1. Ouvrez le fichier `articlebio.txt`.
2. Dans cet article, retrouvez toutes les occurrences du mot *bioinformatics*
3. Remplacez toutes les occurrences de ce mot par *BIOINFORMATICS*

Emacs pour les programmeurs

L'intérêt d'emacs, en dehors de l'exercice intellectuel qu'il peut représenter, est l'ensemble des fonctionnalités qu'il peut offrir pour la programmation.

Exercice 11.

1. Créez un nouveau fichier `hello.c` dans le répertoire `tp2`. Que remarquez-vous dans la barre d'information ?
Recopiez le programme C suivant :

```
# include <stdio.h>
int
main()
{
    printf("Hello World!\n");
    return 0;
}
```
2. Divisez la fenêtre en deux et ouvrez le texte `articlebio.txt`. A quoi sert la touche `[tab]` dans les différentes fenêtre ? Fermez la fenêtre correspondant à `articlebio.txt`
3. Parcourez votre programme du début à la fin en pressant la touche `[tab]` pour chaque ligne. Si votre programme est correct, il doit maintenant être bien indenté.

4. Sauvegardez le fichier, puis, dans un terminal, tapez la commande `gcc hello.c` pour compiler le programme. Corrigez les erreurs jusqu'à ce qu'il n'y en ait plus. N'oubliez pas de sauvegarder.
5. Dans un terminal tapez la commande `./a.out`.