

## TP1 - Prise en main de l'environnement Unix.

Les sujets de TP sont disponibles à l'adresse

<http://www.pps.jussieu.fr/~tasson/enseignement/bioinfo/>

Les documents de cours sont ou seront disponibles sur

<http://www.pps.jussieu.fr/~jch/enseignements/bioinfo/>

### Ouverture de session

Lorsque vous allumez le Neoware, une première fenêtre de connexion apparaît, cliquez sur la flèche verte **connect**. Une deuxième fenêtre de connexion apparaît avec une liste de serveurs, choisissez **monjetas** et cliquez sur **ok**. Si **monjetas** n'est pas dans la liste, vous devez taper dans la zone de texte en bas à gauche de la fenêtre.

Vous devez alors saisir votre identifiant (*login*) et votre mot de passe (*password*) qui vont vous être distribués.

### Interface graphique

Une fois vos nom d'utilisateur et mot de passe vérifiés, le gestionnaire de bureau apparaît. Le menu principal permet de lancer des applications, d'accéder aux outils de paramétrage du système ou encore de fermer la session ou d'éteindre l'ordinateur.

#### Exercice 1 (Exploration).

1. Dans le menu principal, trouvez et lancez l'éditeur de texte *emacs*, le navigateur web *firefox* et un terminal *shell*. Déplacez, modifiez la taille des fenêtres des applications lancées.
2. Dans la fenêtre *emacs*, parcourez les menus **file**, **edit** et **buffer** et essayez de deviner l'utilité de chacune des fonctions.
3. Avec le bouton de droit de la souris, vous pouvez faire apparaître différentes commandes selon l'endroit où vous avez cliqué. Essayez cette fonctionnalité.

#### Exercice 2 (Gestionnaire de fichiers).

1. Ouvrez le gestionnaire de fichiers et explorez l'arborescence du système en repérant le répertoire racine puis votre répertoire personnel.
2. Créez un répertoire appelé **essai** dans votre répertoire personnel et renommez-le en **test**.

### Exercice 3 (Editer un fichier texte).

1. Lancez l'éditeur de texte `emacs` et créez un nouveau fichier.
2. Créez un petit texte vous présentant et sauvegardez le sous le nom `presentation.txt`. Ce texte ne doit pas comporter plus de 4 lignes et il doit faire apparaître vos nom, prénom, âge, parcours à l'université.

### Exercice 4 (navigateur).

Lancez le navigateur web `firefox` et consultez les pages

`http://www.pps.jussieu.fr/~tasson/enseignement/bioinfo/`

Les documents de cours sont disponibles sur

`http://www.pps.jussieu.fr/~jch/enseignements`

## Shell

L'interface graphique permet d'interagir avec le système par l'intermédiaire de menus et de clics. Une autre façon d'interagir avec le système est d'utiliser un *terminal* (ou *console*) dans lequel vous pourrez taper des commandes. Le programme qui exécute vos commandes s'appelle le *shell*.

Le shell vous indique qu'il est prêt en affichant en début de ligne une *invite* (ou *prompt*) en général le caractère dollar (\$) ou supérieur (>). Vous lui indiquez que vous avez fini de taper une commande en appuyant sur la touche [entree].

### Exercice 5 (echo).

1. Tapez dans un terminal ce qui suit :  
`echo "toto"`  
Que se passe-t-il? Quelle touche devez-vous presser pour interagir avec le terminal?
2. Tapez (et validez)  
`echo -n "toto"`  
Que se passe-t-il?

*En résumé.*

- `echo` est une commande unix
- `"toto"` est un paramètre de la commande
- [enter] permet de valider la commande
- `-n` est une option de la commande

On peut distinguer plusieurs catégories de commandes :

- des commandes simples permettant de manipuler des fichiers sur le disque dur et qui affichent leur résultat directement sur le shell (`echo`, `ls`, `cd`, `cp`, ...); ces commandes utilisent directement le shell pour leurs interactions et ne peuvent donc pas être exécutées en dehors du terminal;

- des commandes plus sophistiquées qui n'utilisent pas le shell (`konqueror`, `emacs`,...); ces programmes peuvent également être exécutés à l'aide d'une icône de l'interface graphique;
- des commandes internes au shell (*built-in commands*) qui permettent d'utiliser certaines fonctionnalités offertes par le shell (`alias`, `exit`,...).

**Notations :** Dans la suite des tps on utilisera toujours

- la police de caractère de machine à écrire pour décrire les `commandes`;
- la police de caractère de machine à écrire italique pour représenter une partie de commande à *remplacer*;
- les crochets pour représenter les touches clavier (par exemple `[espace]`).

## Premières commandes

### Mot de passe

#### Exercice 6.

Pour commencer, il vous faut changer de mot de passe. Pour cela, tapez la commande `passwd`. Et répondez aux questions du shell.

### Mail

Votre adresse mail à l'ufr d'informatique est la suivante :

`login@bioinf.informatique.univ-paris-diderot.fr`.

Pour consulter vos mails et en envoyer, vous pouvez utiliser la commande `mail` dont vous trouverez un descriptif dans la documentation fournie en cours.

#### Exercice 7.

Envoyez à l'adresse `tasson@pps.jussieu.fr` votre texte `presentation.txt` à l'aide de la commande `mail` et d'un copier-coller qui se fait en sélectionnant le texte avec la souris puis en cliquant simultanément sur les deux boutons de la souris sur l'endroit où l'on veut coller.

### Qui, quoi, comment

Dans ce paragraphe, nous allons utiliser les commandes `man`, `whoami`, `who w` (pour `what`). Consultez l'aide mémoire Unix distribué en cours pour connaître leur syntaxe et leur fonction.

`man` ouvre un programme qui permet de lire la page de documentation. Vous pouvez faire défiler le texte en utilisant la touche `[espace]`. Pour sortir, pressez la touche `[q]`

#### Exercice 8.

1. Testez les commandes `who`, `whoami` et `w` et observez les informations affichées.
2. Ouvrez la page de manuel de la commande `w`. Quels sont les arguments et les options possible de `w`.

## L'historique

Il arrive parfois que l'on ait à utiliser une même commande plusieurs fois, ou que l'on souhaite corriger une commande tapée précédemment. Sur la plupart des *shells*, il existe un ensemble de raccourcis permettant d'accéder à l'historique des commandes.

Les flèches [ $\uparrow$ ] et [ $\downarrow$ ], permettent de faire défiler les commandes précédentes de la plus récente à la plus ancienne.

Avec les flèches [ $\rightarrow$ ] et [ $\leftarrow$ ] vous pouvez corriger un détail de chaque ligne de commande, ce qui peut être plus rapide que de retaper une longue commande presque identique.

**Exercice 9** (Naviguer dans l'historique). Utiliser les flèches pour trouver à quoi servent les commande `who` et `whoami` sans retaper la commande `man` ?

## Fichiers et répertoires

Il suffit de quelques commandes : `pwd`, `cd`, `ls` pour être en mesure d'effectuer toutes les opérations utiles sur le système de fichier (parcours , copie, déplacement,...) par la ligne de commande.

Consultez l'aide mémoire Unix et les pages du manuel en ligne pour connaître la syntaxe et la fonction de chaque commande.

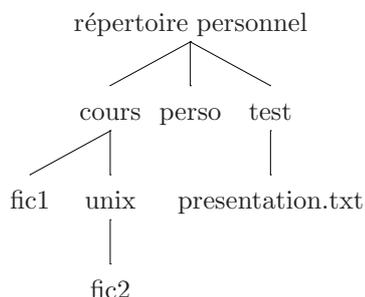
## Arborescence

**Exercice 10** (Tour du propriétaire).

En utilisant les trois commandes ci-dessus, visitez votre répertoire personnel et promenez-vous dans l'arborescence en affichant le contenu de quelques répertoires. Allez aussi visiter les répertoires de vos voisins. Si vous êtes perdu(e), tapez la commande `cd` ou `cd ~`. Quel répertoire est désigné par `~` ?

On pourra utiliser les commandes `mkdir`, `touch` dont on trouvera une description dans le manuel distribué en cours.

**Exercice 11** (Reforestation). A partir de votre répertoire personnel et à l'aide du *shell* créer l'arborescence suivante :



Pour vérifier que vous ne vous êtes pas trompé(e), tapez à partir de votre fichier personnel la commande `ls -R`. A quoi sert l'option `-R`? N'oubliez pas d'utiliser le manuel en ligne (`man`).

Dans les exercices suivants, on aura besoin des commandes suivantes : `cp`, `rm`, `rmdir`, et `mv`. Cherchez dans l'aide mémoire unix leur signification et syntaxe.

**Exercice 12** (Manipulation de fichiers).

1. Dans votre répertoire personnel et à l'aide du *shell* créez une copie de `presentation.txt` appelée `.feuille` (n'oubliez pas le point !).  
Supprimez le fichier `presentation.txt`.
2. Affichez le contenu de votre répertoire personnel. Que remarquez-vous? Réessayez en ajoutant l'option `-a` à `ls`. Concluez en vous aidant éventuellement de la documentation en ligne.

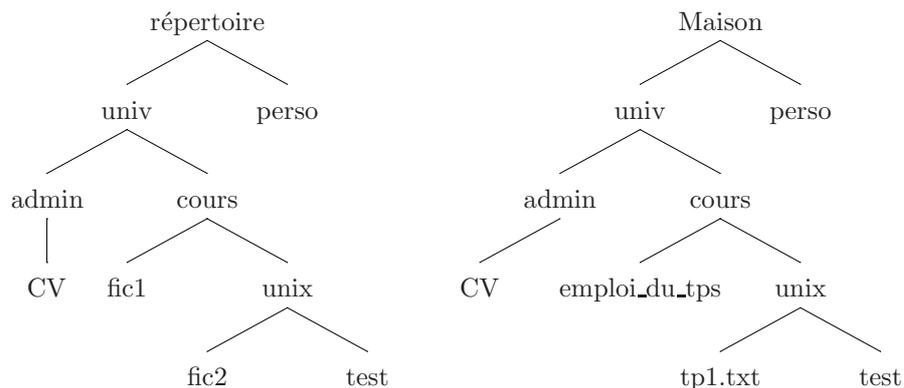
**Exercice 13** (Manipulation de répertoires).

1. A partir de votre répertoire personnel essayez de copier le répertoire `cours`. A l'aide de la commande `man`, trouvez l'option qui vous permet de copier ce répertoire.
2. A partir de votre répertoire personnel, essayez de détruire le répertoire `test` puis le répertoire `cours`. Que se passe-t-il? Résoudre ce problème.
3. Afficher le contenu de votre répertoire personnel. Que remarquez-vous? Réessayez en ajoutant l'option `-a` à `ls`. Concluez en vous aidant éventuellement de la documentation en ligne.
4. Quels répertoires sont désignés par `.` et `...`

**Exercice 14** (Déplacements).

Expérimentez la commande `mv` pour transformer votre arborescence vers l'arborescence ci-dessous à gauche.

Puis, toujours avec la commande `mv`, renommez vos fichiers pour obtenir l'arborescence de droite.



## Recherche de nouvelles options

Les exercices suivant utilisent la documentation en ligne obtenue avec la commande `man`. Prenez l'habitude de le consulter pour vous remémorer la syntaxe des commandes ou de leurs options.

### Exercice 15 (Promotion de `rm`).

En utilisant `rm` de façon récursive, on peut effacer un répertoire et son contenu en une commande.

1. Quelle est cette option ? Appliquez la sur le répertoire `rep3`.
2. Par défaut, `rm` ne demande pas de confirmation ce qui peut s'avérer dangereux ! Trouvez l'option qui permet de remédier à ce problème.

### Exercice 16 (Différents listing).

Par défaut, le résultat de la commande `ls` est minimal.

1. Trouvez l'option permettant d'afficher tous les fichiers, y compris les fichiers cachés.
2. Trouvez l'option permettant d'afficher le maximum d'information sur les fichiers listés.

## Visualiser le contenu de fichiers

Il existe plusieurs commandes pour visualiser des fichiers par exemple : `cat`, `less` (vous pouvez faire défiler le texte en utilisant les flèches [`↑`] et [`↓`] et pour sortir, vous devez presser la touche [`q`]).

### Exercice 17.

Le *chemin relatif* d'un fichier est son chemin par rapport au répertoire dans lequel vous vous trouvez.

Le *chemin absolu* est le chemin par rapport à la racine de l'arborescence.

1. Quel est le chemin absolu du répertoire dans lequel vous vous trouvez ? indice, nous avons déjà utilisé cette commande.
2. En utilisant les différents chemins et différentes manières de visualiser un texte, trouvez et testez quatre commandes permettant de lire le fichier `presentation.txt`.

## Se déconnecter

Pour quitter l'environnement graphique, il faut d'abord avoir quitté toutes les applications ([`ctrl d`] ou `exit` dans un terminal). Ensuite, vous pouvez éteindre l'ordinateur à partir du menu de l'interface graphique.

**NB :** Unix, comme tout système moderne, doit être arrêté proprement et pas en éteignant physiquement la machine.

## Pour aller plus loin

Cette partie du TP est facultative. Vous ne devez la commencer que si vous avez terminé tous les exercices précédents. Il est par contre recommandé de la faire chez vous et de poser des questions au prochain TP.

### Aide pour taper les commandes

! *bla* : réexécute la dernière commande contenant la chaîne de caractère *bla* (pratique lorsque l'on veut refaire une commande âgée sans utiliser les flèches).

### Raccourcis d'édition de commandes

Il existe quelques raccourcis qui peuvent être bien utiles lorsque vous modifiez des commandes un peu longues.

[**ctrl+a**] : ramène en début de ligne.

[**ctrl+e**] : amène en fin de ligne.

[**ctrl+k**] : efface les caractères jusqu'à la fin de la ligne.

[**ctrl+u**] : efface les caractères jusqu'au début de la ligne.

### Exercice 18.

Refaire les exercices 11 et 14.

### Manipuler des ensembles de fichiers

Il existe un certain nombre de symboles spéciaux appelés *Jockers* (ou *wild-cards*), qui permettent de désigner plusieurs choses à la fois. Voici les principaux :

\* : Dans un nom de fichier, représente n'importe quelle chaîne de caractère y compris la chaîne vide. Par exemple **truc\*** désigne *truc*, *trucage*, *trucidule* mais pas *struck*.

? : Dans un nom de fichier, représente exactement un caractère quelconque. Par exemple, **tp?** désigne *tp1*, *tp2*, *tps* mais pas *tp10*.

### Exercice 19.

1. Affichez dans le répertoire **/usr/bin/** tous les fichiers qui commencent par **e** et contiennent au moins 6 caractères.
2. Affichez tous les fichiers cachés de votre répertoire personnel

## Retrouver ses fichiers

`find rep -name "nom"` : retrouve des fichiers situés dans le répertoire `rep` ou dans un de ses sous-répertoires. Dans `nom` les symboles `*` et `?` sont interprétés comme des Jockers.

Par exemple, `find . -name "*tp*"` permet de retrouver tous les fichiers et sous-répertoires du répertoire courant dont le nom contient la chaîne `tp`.

### Exercice 20.

1. Affichez tous les fichiers et sous-répertoires de votre répertoire principal.
2. En utilisant l'option `-type` n'affichez que les fichiers puis que les répertoires.
3. En utilisant l'option `-newer`, affichez tous les fichiers de votre répertoire ont été créés après le début du `tp`.

`grep "expr" fich1 fich2` : affiche toutes les lignes contenues dans les fichiers indiqués contenant l'expression `expr`. Dans `expr`, certains caractères ont une signification particulière (différente de celle des jockers!) :

- `.` : signifie un seul caractère quelconque
- `*` : autant de fois que l'on veut l'expression qui se trouve avant.
- `^` : au début de la ligne
- `$` : à la fin de la ligne

Par exemple, `^1*$` désigne les lignes composées uniquement de `1` et des lignes vides. Si on veut supprimer les listes vides, il faut faire `^11*$`. L'expression `ab(ab\)*` permet de trouver les lignes de la forme `abababababa...`

### Exercice 21.

1. Quelle expression utiliser pour trouver une ligne contenant un groupe de `a` par multiples de trois?
2. Quels sont les fichiers dans votre répertoire personnel contenant votre nom?
3. En vous aidant de la commande `man`, trouvez l'option de `grep` permettant d'afficher les fichiers ne contenant pas une certaine expression.
4. Quelle option permet de retrouver une expression sans tenir compte des majuscules ou minuscules?

## Personnaliser son environnement

### Alias

Certaines commandes sont longues à taper, notamment lorsqu'il y a des options. La commande `alias` permet de remédier à ce problème en vous permettant de créer vos propres commandes.

`alias` : Utilisée sans argument, liste tous les raccourcis.

`alias new_cmd = cmd_complete : new_cmd` est le nouveau nom de la commande et `cmd_complete` est le nom de l'ancienne commande avec ses options (il faut placer les espaces éventuelles entre guillemets).

### Exercice 22.

1. Créez une nouvelle commande `la` qui affiche tous les fichiers y compris les fichiers cachés.
2. Changez le fonctionnement de la commande `rm` pour qu'elle demande toujours confirmation.
3. Fermez le terminal et ouvrez-en un autre. Testez les commandes que vous venez de définir, que constatez vous ?

### Fichiers de configuration

Pour remédier au problème soulevé à la fin de l'exercice précédent, il existe un moyen de stocker ce genre de paramètre une fois pour toutes. On utilise les *fichiers de configurations* liés au shell. Pour le shell `bash`, ces fichiers sont `.bash_profile` et `.bashrc`. Ce sont des fichiers texte contenant des commandes qui sont lues au moment de l'ouverture d'un terminal. Toute modification de ces fichiers ne sera donc pris en compte que si on relance le shell.

### Exercice 23.

1. Editez le fichier `.bashrc` situé dans votre répertoire maison avec `emacs`.
2. Rajoutez les commandes de l'exercice précédent.
3. Testez si ces changements son effectifs (pensez à rouvrir un terminal).

Sur la page <http://www.pps.jussieu.fr/~tasson/enseignement/bioinfo/> vous trouverez les fichiers `.bashrc` et `.bash_profile` qui contiennent quelques commandes pour que votre terminal ait meilleure mine. Ouvrez ces fichiers et essayez de comprendre ce qu'ils veulent dire.