

# Méthodologie de la programmation

## TP 0 : Terminal, shell, éditeur de texte

Dans ce TP :

- Se familiariser avec l'environnement informatique des salles de TP (terminal, shell, éditeur de texte).

### Exercice 0.

Un *terminal* (ou *émulateur de terminal*, ou *console*) est un logiciel qui sert de point d'accès de communication entre l'humain et la machine. Le plus souvent, on entend par terminal une fenêtre dans laquelle est lancé un *shell*.

Un *shell* est un interpréteur de commande. On lui envoie des commandes au clavier, le plus souvent un programme à exécuter et les arguments à lui donner en entrée, et le shell exécute la commande et sert au programme à la fois d'*entrée standard* et de *sortie standard* (c'est à dire que par défaut le programme va lire ce qu'on tape dans le shell et ce qu'il va écrire sera affiché dans le shell).

Dans cet exercice on va voir comment utiliser quelques commandes de base pour explorer l'*arborescence* de *répertoires* et de *fichiers*.

1. → Ouvrir un terminal.
2. Le programme **echo** affiche simplement sur sa sortie standard ce qu'on lui a donné en argument.  
→ Exécuter la commande **echo "coucou"**.
3. Par défaut, vous êtes dans votre *répertoire maison*, c'est la *racine* de la partie du système de fichier qui appartient à votre compte.

On appelle le répertoire courant le *répertoire de travail*, ou *working directory* en anglais. La commande **pwd** (pour "Print working Directory") affiche le *chemin* de ce répertoire.

Un *chemin* contient des noms de répertoire séparés par des */*, chaque */* veut dire que l'on entre dans un sous-répertoire. Dans chaque répertoire il y a deux répertoires cachés :

- **.** est le répertoire lui-même, et
- **..** est le répertoire parent, c'est à dire celui qui est juste au dessus dans l'*arborescence*.

Le dernier élément d'un chemin peut être un fichier qui n'est pas un répertoire (comme par exemple un fichier texte, un fichier audio, etc.).

Si un chemin commence par un */*, on dit qu'il est *absolu*, c'est à dire qu'il commence à la racine du système de fichier.

Sinon, on dit qu'il est *relatif*, et il commence dans le répertoire de travail.

- (a) → Afficher le chemin de votre répertoire de travail avec la commande **pwd**.
- (b) → Quel est le chemin absolu du répertoire parent de votre répertoire maison ?

4. Il existe un raccourci pour l'adresse de votre répertoire maison, c'est **~** (tilde). C'est ce que vous voyez dans votre *invite de commande* (ou *prompt*).

Par défaut, le prompt affiche votre login, puis un **@**, puis le nom de la machine sur laquelle vous êtes connecté, puis un **:**, puis le répertoire de travail, et enfin un **\$** qui signifie que vous pouvez entrer une commande.

→ Vérifier que la commande **echo ~** donne bien la même chose que **pwd**.

5. Pour afficher la liste des fichiers présent dans un répertoire, la commande est **ls chemin-du-répertoire** (pour "List"). Par défaut, **ls** liste les fichiers du répertoire de travail.

La commande **ls** accepte en argument des *options*, par exemple :

- **--all** ou **-a** permet de lister aussi les fichiers cachés (ceux qui commencent par un **.**),
- **-l** donne plein d'informations en plus sur chaque fichier (on ne s'attardera pas sur leur signification).

→ Afficher la liste des fichiers présent dans votre répertoire maison, puis afficher cette même liste mais avec les fichiers cachés.

6. Pour créer un répertoire, on utilise la commande **mkdir chemin-du-répertoire** (pour "Make Directory").  
→ Créer un répertoire **mp** dans lequel on mettra tout ce qui est lié à ce cours. Vérifier avec **ls** que le répertoire est bien là.

7. On peut "aller" dans un répertoire avec la commande **cd chemin-du-répertoire** (pour "change Directory").

- (a) → Aller dans le répertoire **mp**.
- (b) → Afficher tout le contenu (y compris caché) du répertoire **mp**. Quelle commande permet de revenir dans le répertoire parent ?

- (c) → Dans le répertoire `mp`, créer un répertoire pour le travail de ce TP (vous pouvez l'appeler comme vous voulez, par exemple `tp1`) puis aller dedans.
8. La commande `rmdir chemin-du-répertoire` (pour "REMOVE DIRECTORY") permet de supprimer un dossier, mais refuse par sécurité de le faire si celui-ci n'est pas vide.  
→ Créer un dossier `foo` et dedans un dossier `bar`, puis tenter de supprimer `foo` directement, puis faire ce qu'il faut pour supprimer `foo`.

## Exercice 1.

Création et édition de fichiers texte.

Processus en arrière plan.

Manuel.

1. Pour créer un fichier, il suffit d'ouvrir un fichier qui n'existe pas. Par exemple, on peut lancer l'éditeur de texte Mousepad et lui demander d'ouvrir le fichier `essai.txt` avec la commande `mousepad essai.txt`.  
Avant de poursuivre, merci de quitter complètement Mousepad (fermer toutes les instances ouvertes du logiciel).
  - (a) → Lancer Mousepad sur le fichier `essai.txt`, et écrire par exemple "coucou" dans le fichier, et l'enregistrer.
  - (b) → Revenir sur le terminal sans quitter Mousepad, que constatez-vous?
  - (c) → Quitter Mousepad et revenir sur le terminal.
2. Pour lancer une commande en *arrière plan* on peut la suffixer avec `&`.
  - (a) → Relancer Mousepad sur le même fichier mais cette fois-ci en arrière plan.
  - (b) → Revenir sur le terminal sans quitter Mousepad, que constatez-vous?
3. On peut afficher le contenu d'un fichier sur la sortie standard avec la commande `cat chemin-du-fichier` (pour "CATenate" (parce qu'elle peut prendre plusieurs fichiers d'un coup)).  
→ Afficher le contenu de `essai.txt`, modifier le dans mousepad, puis le réafficher.
4. On peut supprimer des fichiers avec la commande `rm chemin-du-fichier` (pour "REMOVE").  
**Attention** la suppression est définitive, ça ne va pas dans une "corbeille".  
→ Supprimer le fichier `essai.txt`.
5. Pour avoir de l'aide sur une commande, il est possible de consulter son manuel d'utilisation avec la commande `man commande` (pour "MANual").  
→ Consulter rapidement les manuels de `echo`, `pwd`, `ls`, `mkdir`, `cd`, `mousepad`, `cat`, `rm`, et `man`.