

## TD1 : Gestion des fichiers

### Objectifs :

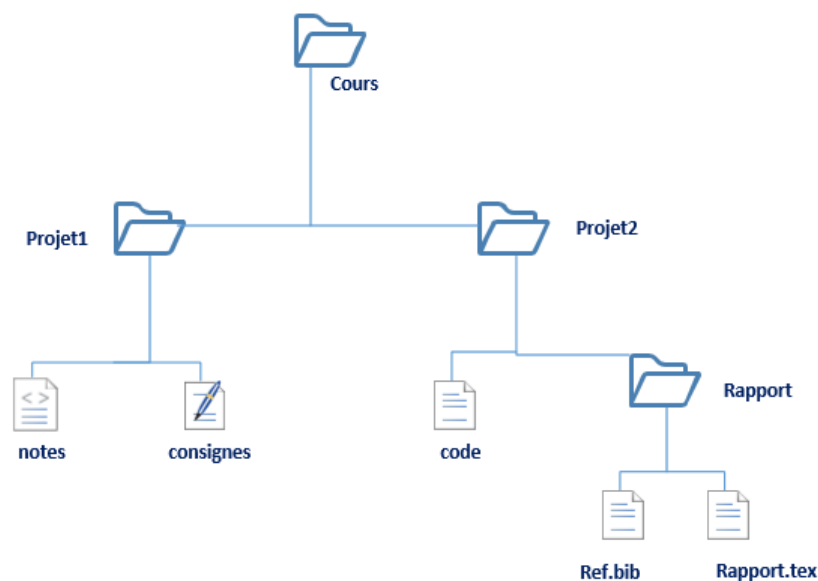
- Découvrir son environnement de travail
- Connaître la structure arborescente d'UNIX et les principaux répertoires
- Maîtriser les commandes élémentaires de gestion des fichiers et des répertoires

### 1. Exploration de l'arborescence Linux

1. Connectez-vous en utilisant votre login et votre mot de passe
2. Quel est votre répertoire personnel?
3. Lancer, dans l'ordre, les lignes de commandes suivantes, puis interpréter les résultats:
  - `$pwd`
  - `$cd ..`
  - `$cd /`
  - `$pwd`
  - `$cd $HOME`
  - `$pwd`
4. Rendre le répertoire `/usr` votre répertoire courant de travail en utilisant un chemin absolu (vérifier qu'on est bien dans ce répertoire)
5. Revenir à votre répertoire personnel (répertoire de connexion) et refaire la question 4 en utilisant un chemin relatif.

### 2. Manipulation des fichiers et des répertoires

1. Créer l'arborescence suivante au niveau de votre répertoire personnel, en notant la suite des commandes que vous avez utilisées.



Les fichiers sont vides.

2. Déplacer le fichier `ref.tex` dans `projet1`.

3. Ajouter du contenu au fichier ref.bib.
4. Créer un lien symbolique (Soft Link) sur le fichier notes nommé soft\_notes.
5. Créer un lien hard (Hard Link) sur le fichier notes nommé hard\_notes
6. Afficher et noter le numéro d'inode des fichiers notes, soft\_notes, hard\_notes.  
Interpréter le résultat en spécifiant la différence entre la notion de lien soft/hard de point de vu allocation de numéro d'inode par le système.
7. Où est ce qu'on peut afficher le nombre de lien Hard pour un fichier.
8. Modifier le contenu du fichier hard\_notes. Afficher le contenu des 3 fichiers hard\_notes, soft\_notes et notes. Qu'est-ce que vous constatez ?
9. Supprimer le fichier notes. Afficher le contenu des deux fichiers hard\_notes, soft\_notes.
10. Créer une copie du fichier « code » nommée « programme.py ».
11. Renommer le fichier code en code.new
12. Supprimer le répertoire rapport.

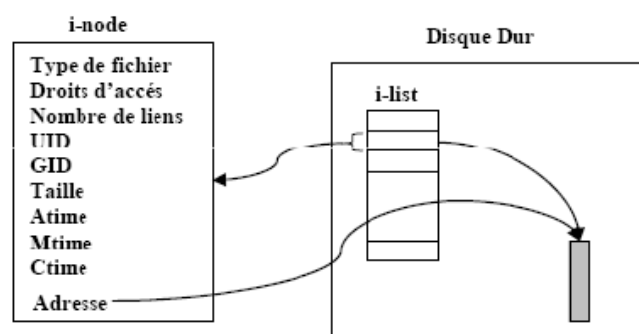
### 3. Annexe

#### A. Table d'inodes

Un **inode** est une structure de données contenant les informations décrivant et représentant un fichier. Ces informations sont appelées des **attributs**. Chaque fichier dispose d'un numéro d'inode (i-number). Tous les inodes sont présents au sein d'une table d'inodes (i-list).

Un fichier ne peut avoir qu'un seul inode. Un inode est unique au sein d'un seul système de fichiers. Chaque système de fichiers dispose d'une table d'inodes indépendante.

Cependant si deux noms de fichiers peuvent se voir associé le même numéro d'inode au sein d'un même système de fichiers, ces deux noms représentent alors un seul et même fichier.



#### Contenu

Le contenu d'un inode varie d'un système de fichiers à un autre, mais la norme POSIX impose que chacun d'eux dispose au moins des attributs suivants pour chaque fichier :

- sa taille
- l'identifiant du périphérique le contenant
- son propriétaire
- son groupe
- son numéro d'inode
- son mode (ses droits) d'accès
- sa date de dernière modification d'inode (change time), à ne pas confondre avec la date de création
- sa date de dernière modification de contenu (modification time)
- sa date de dernier accès (access time)
- un compteur de hard links (liens physiques ou durs, voir plus loin)

On peut obtenir quelques informations sur un inode avec la commande **stat**.

Un inode ne contient pas le nom du fichier ni d'informations sur la date de création d'un fichier.

## B. Les interfaces

Sous Linux, on peut facilement basculer du mode graphique vers l'interface texte avec l'association de touches CTRL+ALT+F1, puis revenir en mode graphique avec CTRL+ALT+F7. Chaque combinaison de touches CTRL+ALT+F1, CTRL+ALT+F2, etc. ouvre une nouvelle en mode texte. Seule l'association CTRL+ALT+F7 bascule vers une console graphique.