

---

# Systemes et reseaux

## TP 1 : Boîte à outils

---

### Exercice 0.

Mise en place

1. Le projet qui vous est fourni implémente une architecture de boîte à outils, où chaque outil est une sous-commande du programme principal **outils**.
2. Un **Makefile** que vous n'aurez a priori pas besoin de modifier est présent dans les fichiers du projet.  
→ Compilez le projet à l'aide du **Makefile**.
3. En l'état, seuls deux outils sont disponibles : **help** et **cat**.
  - L'outil **help** est implémenté directement dans le fichier principal **outils.c**.
  - L'outil **cat** est implémenté dans son propre fichier **cat.c**.  
Dans le fichier principal on retrouve la déclaration de la fonction qui implémente l'outil (ligne 7) et l'enregistrement de cette fonction comme sous-commande (ligne 14).  
→ Que font ces deux outils ? À quelles conditions sont-ils appelés ?

### Exercice 1.

Implémentation d'un premier outil.

1. Notre premier outil additionnel sera **pwd**.
  - (a) Que fait cette commande (la vraie) ?
  - (b) Quelle fonction de la bibliothèque standard nous permet de récupérer cette information ?
  - (c) Dans quel entête de la bibliothèque standard est-elle définie ?
  - (d) Comment cette fonction s'utilise-t-elle ?
2. On va implémenter notre commande dans un nouveau fichier **pwd.c** dans lequel on définira une fonction **outil\_pwd**.  
→ Pour s'intégrer à notre boîte à outil, quel devra-t-être le prototype de cette fonction ?
3. → Dans le fichier principal, déclarez et enregistrez la nouvelle commande dans la boîte à outils.
4. Dans le fichier **pwd.c**, implémentez le nouvel outil.
5. Il est important de régulièrement vérifier que votre code compile sans erreur ni avertissement.  
→ Recompilez le projet et testez que votre nouvelle commande fonctionne.

### Exercice 2.

Implémentation d'un deuxième outil.

1. Notre deuxième outil sera une version simplifiée de **ls**.  
Cet outil affiche la liste des fichiers présents dans le répertoire courant, avec un fichier par ligne, en ignorant les fichiers cachés (dont le nom commencent par un point).  
Si son premier argument est "**-l**", il affiche également avant le nom de chaque fichier les droits de l'utilisateur en lecture (**r** ou **-**), écriture (**w** ou **-**), et exécution (**x** ou **-**) pour ce fichier.  
→ De quelles fonctions allez-vous avoir besoin ? Dans quels entêtes de la bibliothèque standard sont-elles déclarées ?
2. → Implémentez l'outil **ls**.

### Exercice 3.

Compléter notre boîte à outil.

1. Notre outil **rm** prend une liste de chemins en argument et supprime chacun des fichiers ciblés par ces chemins.  
→ Implémentez l'outil **rm**.
2. Notre outil **cp** prend deux chemins en argument et copie le fichier ciblé par le premier chemin à l'endroit pointé par le second chemin si aucun fichier n'existe déjà à cet endroit là (sinon on affiche une erreur).  
→ Implémentez l'outil **cp**
3. → Pourquoi est-il impossible d'implémenter l'outil **cd** ?