

Programmation système

TP 3 – Implémentation de `ls`

Exercice 1: Un `ls` simplifié

1. Si vous ne les avez pas encore fait, faites les exercices bonus 2 et 3 de la feuille précédente. Vous obtiendrez un `ls` qui accepte l'option `-R` pour fonctionner de façon récursive.

Indication : Pour l'instant, il n'est pas nécessaire de gérer la récursivité comme le "vrai" `ls`, qui liste d'abord le contenu du répertoire puis descend dans les sous-répertoires. Vous pouvez descendre dans les répertoires au fur et à mesure que vous les rencontrez.

2. Modifiez votre programme afin que l'affichage soit paramétrable :
 - Sans options, l'affichage sera similaire à celui de `ls -l` ("ls moins un"). Autrement dit, vous n'avez pas besoin d'implémenter l'affichage en plusieurs colonnes.
 - Les options `-l` ("moins ell") et `-n` auront le même effet que pour le `ls` du système.

Indications :

- Utilisez toujours `getopt()` pour l'analyse de la ligne de commande.
 - Pour convertir les UID et GID en noms, utilisez `getpwuid()` et `getgrgid()` ; pour convertir la date, utilisez `ctime()` ou une fonction similaire.
 - Pour l'instant, il n'est pas nécessaire de trier les répertoires, ni d'afficher leur taille totale.
3. Modifiez le programme pour qu'il utilise `scandir()` à la place d'`opendir()` et `readdir()`.
 4. Triez les répertoires par ordre alphabétique (comme `ls`). Indice : il y a un paramètre `compar` dans `scandir()` qui permet de le faire très facilement.

S'il vous reste du temps:

Exercice bonus 1: Un `ls` presque complet

1. Écrivez une fonction `isDir(const struct dirent *)` qui indique si la structure passée est un répertoire ou non. Puis, utilisez cette fonction pour que votre `ls`, lorsqu'il fonctionne en mode récursif, liste d'abord le contenu du répertoire, puis descende dans les sous-répertoires.
2. Bravo, vous avez un `ls` qui marche ! Si vous êtes arrivés jusque-là, implémentez aussi les options `-a`, `-d`, `-i`, `-s`. Pour comparer votre `ls` avec le `ls` "normal", utilisez ce dernier sans alias (i.e., `/bin/ls` au lieu de `ls`) et toujours avec l'option `-l` ("moins un").