

Programmation système

TP 2 – Lecture, écriture, et métadonnées

Exercice 1: cat

1. En utilisant les appels système `read()` et `write`, implémentez en C une version simplifiée de la commande `cat` qui lit ce qu'on lui donne sur son entrée standard (descripteur 0) et le recopie sur sa sortie standard (descripteur 1). Attention, il ne faut pas utiliser les fonctions de la bibliothèque `stdio` dans cet exercice.
2. Modifiez votre programme pour qu'il fonctionne comme suit :
 - S'il n'y a pas d'arguments, il se comporte comme avant.
 - Sinon, il affiche l'un après l'autre le contenu de tous les fichiers donnés en argument (exactement comme la commande `cat` du système). Dans ce cas, le programme ignore donc son entrée standard, mais il écrit toujours sur sa sortie standard.

Remarque : ici vous aurez aussi besoin des appels système `open()` et `close()`.

Exercice 2: stat()

Si vous avez fait l'exercice bonus 1 de la feuille précédente, vous pouvez sauter cet exercice.

En utilisant l'appel système `stat()` (cf. `man 2 stat`), écrivez un programme qui affiche le numéro d'inode, la taille, la date de modification, et le type (fichier, répertoire, ou autre) d'un fichier passé en argument. Inutile de mettre en forme la date de modification ; il suffit de l'afficher en l'état (c'est à dire en secondes depuis le premier janvier 1970). Quant au type, affichez juste "f", "d", ou "?" pour indiquer un fichier normal, un répertoire, ou autre chose. Pour connaître le type du fichier, utilisez les macros `S_ISREG` et autres sur le champ `st_mode` de la structure passée à `stat()`. La page `man` de `stat()` contient une référence vers une autre page `man` décrivant ces macros.

Exercice 3: lstat()

Modifiez le programme de l'exercice 2 pour indiquer si le fichier est un lien (par la lettre "l") et donner la destination du lien. Pour cela, utilisez les appels système suivants :

- `lstat()`, pour récupérer les métadonnées du fichier. Attention : contrairement à `lstat()`, `stat()` suit toujours les liens, et n'indiquera jamais si un fichier est un lien ou pas ! Par ailleurs, si on demande à `stat()` de suivre un lien invalide, il renverra une erreur.
- `readlink()`, pour connaître la destination du lien. Prenez le temps de bien lire la page `man` correspondante.

S'il vous reste du temps:

Exercice bonus 1: Un cat amélioré

1. Modifiez votre implémentation de `cat` de l'exercice 1 pour qu'elle accepte l'argument spécial "-", qui représente l'entrée standard. (Cf. la commande `cat` du système.)

2. La page man de `write()` indique que dans certains cas, cet appel système peut écrire moins d'octets que ce qui a été demandé (et ce n'est pas une erreur). Avez-vous pris cela en compte ? Si ce n'est pas le cas, modifiez votre programme en conséquence.
3. Ajoutez l'option `-n` qui affiche les numéros de ligne. (Servez-vous de `getopt()`.)
Remarque : cette dernière partie est un peu plus difficile.

Exercice bonus 2: Un prototype de `ls`

1. En utilisant les fonctions de bibliothèque `opendir()` et `readdir()`, écrivez un programme qui liste le contenu du répertoire courant, en affichant juste le nom de chaque fichier. N'oubliez pas le `closedir()` à la fin.
Attention : il y a deux pages man pour `readdir()` ; utilisez celle de la section 3.
2. Maintenant, affichez aussi les mêmes informations que dans l'exercice 3 (réutilisez le code), en ajoutant le nom de chaque fichier, bien sûr.
3. Modifiez votre programme afin que :
 - s'il n'y a pas d'argument, il fonctionne comme avant,
 - si le premier argument est un fichier, il fonctionne comme le programme de l'exercice 3,
 - si le premier argument est un répertoire, il liste le contenu de ce répertoire,
 - s'il y a plusieurs arguments, il les traite les uns après les autres.*Attention* : il faut concaténer le nom du répertoire et le nom du fichier ; nous conseillons l'utilisation de `snprintf()`.

Exercice bonus 3: Un `ls` récursif

Modifiez le programme de l'exercice bonus 2 afin qu'il accepte l'option "`-R`" pour fonctionner de façon récursive (comme `ls`). Utilisez `getopt()` pour traiter les arguments de la ligne de commande.