Séance de TP N°2
02. lsrecursif - commande ls -lR récursive

Créez votre répertoire de TP *~/l3i6/02.lsrecursif* et placez-vous dessus. Récupérez dans votre répertoire de TP le fichier *Fourniture02.tgz* sur le site de cet enseignement[1] et déballez-le. Cette archive contient :

- un script configure et son ***Makefile*** générique ***Makefile.gen*** ;
- ***printAttributes1.c*** et ***printAttributes2.c*** : deux squelettes à compléter pour afficher en clair les attributs de fichiers ;
- ***printPermissions1.c*** et ***printPermissions2.c*** : deux versions qui affichent en clair les permissions de fichiers ;
- ***lsl.c*** : une version non récursive de la commande *ls -l* qui affiche toutes les informations sur les fichiers donnés en arguments, délivrées par les primitives *stat*, *fstat* et *lstat*.
- ***test-internationalisation***, un script shell de test de l'internationalisation du système Unix,
- ***fourniture.c*** (squelette de ***parcoursRecursif.c***).
- Le ou les fichiers à compléter sont les suivants : ***lsl.c***, ***printAttribute1.c***, ***parcoursRecursif.c***, ***printAttribute2.c***

Version non récursive

Réalisez un programme ***lsl.c*** qui réalise, pour tous les fichiers donnés en argument, un affichage en clair de tous leurs attributs, délivrés par les primitives *stat*, *fstat*, *lstat* dans une structure *stat*. Notez que les commandes Unix **ls** n'affichent jamais d'un seul coup tous les attributs d'un fichier, car il faudrait plusieurs lignes par fichier. L'utilisateur doit donc choisir les informations qu'il désire par des options de la commande **ls** et s'il les veut toutes, s'y prendre en plusieurs fois. Ici, avec la commande **lsl**, l'affichage est complet, mais prend plusieurs lignes.

Pour cela vous compléterez les fichiers ***lsl.c***, ***printAttributes1.c*** en vous aidant du fichier ***printPermissions1.c***. Dans le squelette du texte source du programme ***PrintAttributes1.c*** vous remarquerez que l'on y fait référence au système d'impression internationalisé d'Unix. Dans un premier temps vous ne chercherez pas à réaliser une version internationalisée.

Une fois que c'est fait, exécutez le ***Makefile*** fourni après avoir exécuté le script **configure**. Ce ***Makefile*** construit le programme **lsl** avec les modules ***PrintAttributes1*** qui utilise à son tour ***printPermissions1***. Puis le programme **lsl** est automatiquement exécuté sur le répertoire ***TMP*** et ses fichiers, qui sont créés localement par le ***Makefile***. Un fichier de traces ***lsl.res*** conserve les affichages.

Version non récursive et Internationalisation

Le système d'impression internationalisé de Unix permet d'afficher les messages d'erreurs, les pages de manuels, le format des dates, le symbole monétaire, *etc*, selon la langue choisie par l'utilisateur, grâce à des variables d'environnement comme LANG ou plus spécialisées comme LC_TIME (*cf.* les manuels *setlocale(3)*, *locale(5)*...). Cependant, les traductions des manuels et des messages d'erreurs dans plus d'une centaine de langues différentes, avec des caractères Unicode prend beaucoup de place sur disque. C'est pourquoi la plupart des machines Unix ou Linux ne conservent que les fichiers nécessaires qu'à quelques langues, voire à une seule si c'est l'anglais.

[1] http://deptinfo.unice.fr/~lahire/enseignement/SYSL3

Pour tester le système d'internationalisation sur votre machine, affichez les variables d'environnement qui pilotent le système d'internationalisation, utilisez des commandes d'affichage de dates (comme **date** ou **ls -l**) et des commandes erronées qui affichent des messages d'erreurs, puis changez la valeur de vos variables d'environnement pour d'autres langues et recommencez l'expérience. Le script **test-internationalisation** fourni effectue une telle expérience pour le français, l'anglais et l'allemand.
Ré-exécutez le programme ***lsl*** fourni en changeant la langue pour les dates (variable LC_TIME).

Dans un deuxième temps, compléter la partie concernant l'internationalisation de l'affichage dans le fichier printAttribute1.c. Ensuite testez votre programme en changeant la valeur des variables d'environnement.

Parcours récursif d'une hiérarchie de fichiers

Ecrire le programme ***parcoursRecursif.c*** qui reprend le programme ***lsl.c*** précédent, mais en lui faisant subir plusieurs modifications :

- affichage plus compact des attributs de fichiers, en complétant les modules ***PrintAttributes2*** qui utilisera ***printPermissions2***,
- affichage récursif, grâce à un parcours récursif des répertoires. On prendra comme squelette du programme ***parcoursRecursif.c*** le fichier fourni ***fourniture.c*** que l'on renommera ***parcoursRecursif.c***. Vous devez compléter les corps des trois routines *recursive_doall*, *count_file* et *print_file_name*, partout où il y a des points de suspension. Comme la principale nouveauté de ce programme est le parcours récursif d'une hiérarchie de fichiers, on généralise le parcours, nécessaire pour afficher les attributs de fichiers, à un parcours capable d'exécuter une action quelconque sur chaque fichier parcouru, à la manière de la commande *find(1)* avec l'option *-exec*. On testera la généralité du parcours sur les trois actions décrites ci-après. Pour chaque argument, ce programme affiche le numéro et le nom de l'argument, puis lui applique successivement trois actions, qui agiront sur l'argument et, s'il s'agit d'un répertoire, récursivement sur tous les fichiers au sens large et les répertoires situés en dessous. Ces actions consisteront à :
    - imprimer le nom relatif du fichier au sens large, sur une ligne,
    - imprimer le nom du fichier au sens large et ses attributs, sur une ligne
    - incrémenter un compteur correspondant au type du fichier au sens large : régulier, répertoire, fichier spécial fonctionnant par blocs, fichier spécial fonctionnant par caractères, fichier *FIFO*, lien symbolique, *socket*.

À la fin du traitement de tous les arguments, les compteurs seront imprimés avec les pourcentages de chaque type. Les trois actions seront programmées dans trois routines de nom *print_file_name*, *printAttributes*, *count_files*. Pour ces trois parcours, l'ordre sera celui de la primitive *readdir(3)* sans tri. Les noms de fichiers affichés seront relatifs aux noms des arguments d'appel du programme. Le parcours récursif est programmé dans la routine :

void *recursive_doall* (**char*** *path_name*, **FILE_HANDLER *** *func*) ;

qui exécute l'action *func* pour le fichier *path_name*, puis, si le fichier fourni est un répertoire, *recursive_doall* s'appelle récursivement pour chacun des fichiers du répertoire (lus avec *opendir(3)*, *readdir(3)*, *closedir(3)*, et en sautant les répertoires "**.**" et "**..**"). Notez bien que l'argument *path_name* sert à contrôler l'état courant du parcours récursif et fonctionne comme une pile, en s'agrandissant ou se rétrécissant au fur et à mesure du parcours, selon que l'on descend ou remonte dans la hiérarchie du répertoire.