

TP3 ¹

1 Répertoire de travail courant

Le but de l'exercice est d'écrire une commande `mon_pwd` qui affiche le répertoire courant équivalente à la commande `/bin/pwd` du système.

La fonction `char *getcwd(char *buf, size_t size)`; de la bibliothèque C copie le chemin absolu du répertoire courant dans le tableau `buf` de taille `size`. Si `buf` est `NULL`, la bibliothèque alloue dynamiquement un espace nécessaire pour y stocker le résultat et en retourne alors l'adresse.

Question 1 *Fournir une implantation de la commande `pwd` basée sur une utilisation de `getcwd`.*

La fonction `getcwd` n'est pas toujours directement un appel système mais simplement une fonction écrite en C (qui elle-même fait des appels système). On se propose donc de réécrire la fonction `getcwd`. Cette fonction doit pouvoir être utilisée comme fonction de librairie : elle ne doit donc pas modifier les variables globales du programme ni créer une fuite de mémoire. Cette fonction remonte récursivement dans la hiérarchie jusqu'à la racine du système en recherchant à chaque étape le répertoire courant “ . ” dans le répertoire supérieur “ .. ”.

Question 2 *Écrire une fonction*

```
int equal_node (struct stat st1, struct stat st1);
```

qui teste si deux noeuds de la hiérarchie de fichiers sont identiques. Deux noeuds sont identiques si et seulement si leurs numéros de noeuds et leurs numéros de partition sont égaux.

Question 3 *Écrire une fonction `int is_root(char *dir)`; qui teste si un répertoire est le répertoire racine.*

Question 4 *Écrire une fonction*

```
char* dir_find(char* dir_name, struct stat file_stat);
```

qui prend en arguments un nom de répertoire `dir_name` et une structure `file_stat` de type `stat` associée à un fichier. Cette fonction recherche dans le répertoire `dir_name` si le fichier associé à `file_stat` est présent. Si la fonction trouve le fichier elle retourne son nom, sinon elle retourne le pointeur `NULL`.

1. Des exercices sont repris des cours de Philippe Marquet et Dider Remy.

Pour écrire cette fonction, on utilisera les fonctions `opendir`, `readdir` et `closedir`. Pour ne pas créer une fuite de mémoire, on fera bien attention à refermer le répertoire ouvert, y compris lorsqu'une erreur se produit.

Les informations sur les fichiers, indispensables à la comparaison de noeuds, sont obtenues par l'appel système `lstat` qui prend en argument un chemin (on n'utilise pas `stat` ici car on recherche un chemin directe issu de la racine qui ne traverse pas de liens symboliques).

Question 5 Écrire une fonction `char *mon_getcwd(char *buf, size_t size)` qui se comporte comme `getcwd`. L'algorithme manipule des chemins et des noeuds. On évitera l'utilisation de `chdir` qui affecterait alors le reste du programme.

2 Droits d'accès

La commande Unix `test` permet de tester les droits d'accès (lecture : option `-r`, écriture : option `-w`, exécution : option `-x`) à un fichier donné.

La commande `test` n'affiche rien mais retourne son résultat sous la forme d'une terminaison sur un succès ou un échec. Ce comportement peut par exemple être mis en valeur dans une session telle la suivante :

```
--> test -r /tmp
--> echo $?
0
--> test -w /etc/passwd
--> echo $?
1
```

Le but de l'exercice est de fournir une commande `maccess` qui le même comportement que `test` avec les options `-r`, `-w` et `-x`. Votre commande doit supporter l'option supplémentaire `-v` (verbose) expliquant, en cas d'échec, ce pourquoi l'accès est impossible. Les raisons de l'échec sont les suivantes :

1. Le droit d'accès demandé au fichier n'est pas positionné.
2. Le fichier n'existe pas.
3. Une des composantes du nom de fichier n'est pas un répertoire.
4. Une des composantes du nom du fichier est trop longue.
5. Le nom du fichier est trop long.
6. Le nom de fichier comporte trop de liens symboliques.
7. Autre erreur.

Question 6 Écrire une commande `maccess`. Vous pourrez utiliser l'appel système `access` pour vérifier les permissions d'accès à un fichier.

Question 7 Fournir un exemple de session shell comportant une suite d'invocation de la commande `maccess` produisant l'ensemble des sept erreurs possibles.