

TD UNIX n°1

Processus et thread sous UNIX

1°) Considérons le programme suivant :

```
int main() {  
    printf("A\n");  
    fork();  
    printf("B\n");  
    fork();  
    printf("C\n");  
    exit(EXIT_SUCCESS);  
}
```

Combien de processus y aura-t-il eu quand le programme sera terminé ?

Combien de fois sera affiché le message "A", le message "B" et le message "C" ?

2°) a) On veut écrire un programme qui recherche un élément dans un tableau d'entiers de grande taille (1 million de valeurs). L'idée retenue est d'écrire un programme qui crée un processus fils par fork puis :

- Le processus père explore la 1^{ère} moitié du tableau
- Le processus fils explore la 2^{ème} moitié du tableau

Celui qui trouve affiche "Le père (resp fils) a trouvé en position : x"

On ne se préoccupera pas de l'initialisation du tableau qui aura pu être faite par une lecture dans un fichier par exemple. On prévoira de saisir au clavier la valeur à rechercher avant de créer le processus fils.

Pour éviter les processus zombies on prévoira que le père attende la terminaison de son fils.

b) Modifier ce programme pour qu'il affiche un message quand la valeur n'a pas été trouvée.

3°) On va maintenant reprendre le même problème en utilisant des threads.

Le programme (processus) saisit au clavier la valeur à rechercher comme précédemment puis crée deux threads.

- Le 1^{er} thread explore la 1^{ère} moitié du tableau
- Le 2^{ème} thread explore la 2^{ème} moitié du tableau

Celui qui trouve place la position où il a trouvé dans une variable globale du programme.

Le processus attend la fin des 2 threads puis affiche en quelle position la valeur a été trouvée ou un message indiquant qu'elle n'a pas été trouvée.

On écrira une fonction différente pour chacun des 2 threads.

4°) Les deux threads du 3°) exécutent à peu près le même algorithme seule la position de départ de leur recherche dans le tableau diffère. Faire une version améliorée de ce programme dans laquelle les 2 threads exécutent la même fonction avec un paramètre différent qui désigne le point de départ de la recherche.