Mémoire partagée

Vous vous doutez que sans possibilité de dialoguer entre processus, le principe de l'exécution parallèle est un peu inutile… La seule chose que nous pourrions faire c'est lancer une tache (du style ouvrir une porte ou afficher une fenêtre de choix de fichier) mais une fois la tache lancée, impossible de dialoguer avec elle…

Nous connaissons deux moyens de communiquer entre processus pour l'instant : le père peut donner des infos au fils grâce aux paramètres et le fils peut donner une info au père grâce au code de retour du **`exit`**(). C'est bien maigre comme possibilités… Heureusement, il y a une autre possibilité : La mémoire partagée.

Qu'est ce qu'une mémoire partagée ?

Dans le contexte du matériel informatique, la **mémoire partagée** désigne un large bloc de mémoire vive qui est accédé par différents processeurs dans un système multiprocesseur. Dans notre cas, cela désigne un bloc en mémoire accessible par différents processus.

En pratique, on va considérer ce bloc en mémoire comme un fichier en lecture/écriture. Donc tout d'abord, il faut créer ce fichier :

```
int idShm = shmget (key, sizeof(Float), IPC_CREAT|0666) ;
```

Key est un identifiant entière de cette zone mémoire, le second paramètre est la taille à réserver et le dernier paramètre permet de créer la zone mémoire avec des droits de lecture écriture pour tout le monde.

Ensuite, il faut récupérer le pointeur sur cette mémoire : **shmat()**

```
float * maVar = (float *) shmat (idShm, NULL, 0) ;
```

Maintenant, tout ce qu'on écrit dans cette variable sera lisible par tous les processus !

Il faut faire maintenant attention a ne pas oublier de supprimer ces zones en mémoire car elles resteraient allouées tout le temps ! il faut alors utiliser la fonction : **shmctr()**

```
float * shmctl(int shmid, IPC_RMID, NULL);
```

Si vous oubliez de faire cette opération, il existe deux commandes linux pour vous aider : ipcs et ipcrm. La première permet de connaitre tous les segments mémoire partagée, la seconde permet de les détruire.

Pour pouvoir utiliser ces fonctions, il faut rajouter des librairies !

Exercice 1 :

Faites un programme qui créé une mémoire partagée en mémoire. Dans cette mémoire (un tableau de 10 cases de **char**), vous mettrez « coucou ».

Faites un autre programme qui récupère la valeur de la mémoire partagée et qui l'affiche à l'écran.

Exercice 2 :

Faites un programme qui créé une mémoire partagée (un **int**) et qui créé un fils.

Le père écrit dans cette variable 0 et attend que la valeur soit égale à 1. Quand c'est le cas, il affiche « le père dit 0 ! » et écrit de nouveau zéro… Cette série se répète à l'infini.

Le fils écrit dans cette variable 1 et attend que la valeur soit égale à 0. Quand c'est le cas, il affiche « le fils dit 1 ! » et écrit de nouveau 1… Cette série se répète à l'infini.

Exercice 3 :

Faites un programme qui créé une mémoire partagée (un **int**) et qui incrémente cette valeur toutes les secondes.

Faites un second programme qui affiche la valeur de cette mémoire.

Exercice 4 :

Faites deux programmes : l'un créé une zone mémoire partagée (un **bool**) , l'initialise à **false**, et attend qu'elle soit égale à **true**. L'autre programme utilise la zone de mémoire partagée et écrit la valeur **true**. Cela doit avoir pour effet de débloquer le premier processus.