

Système M1 – TP 11 : Mémoire partagée - threads

Semaine du 7 décembre 2009

Exercice 1 – Comptons avec la mémoire partagée bis

Consultez le code source fourni qui implémente l'exercice 2 du TP9. Le père écrit tous les nombres pairs et le fils nombres impairs dans un segment de mémoire partagée plutôt que dans un fichier.

Modifier le code pour aussi stocker la somme des nombres déjà écrits dans une variable partagée `somme` (on pourrait par exemple la placer au début du segment partagé) .

Exercice 2 – Des lecteurs et des écrivains (du TP9)

On dispose d'un fichier `toto`, que certains processus veulent lire et sur lequel d'autres veulent écrire. On veut implémenter l'idée suivante :

- Les processus écrivains ne peuvent écrire qu'un par un.
- Les processus écrivains ne peuvent écrire que si tous les lecteurs ont fini de lire.
- Les processus lecteurs ne peuvent lire que si aucun écrivain n'est en train d'écrire.

1. On implémentera cette idée au moyen de deux sémaphores, en complétant le code source du fourni.
2. On veut autoriser un nombre maximal de processus lecteurs en même temps. Comment pouvez-vous réaliser ceci ?

Implémenter en complétant le code fourni.

Exercice 3 – Hello Thread

Ecrire un programme qui d'abord lance 3 threads qui affichent "Hello Thread", leur numéro et leur identifiant de thread, puis qui attend leur terminaison et affiche "Bye". Vous utiliserez les primitives `pthread_create`, `pthread_join` et `pthread_self`. Exemple :

```
$ ./hello
Hello! (Thread 1 - 3076119440)
Hello! (Thread 2 - 3067726736)
Hello! (Thread 0 - 3084512144)
Bye! (Thread main - 3084515008)
$
```

A la compilation ajoutez l'option `-pthread`.

Exercice 4 – Lecteurs/écrivains allégés

Reprogrammez les lecteurs/écrivains en utilisant les threads à la place des processus.