

Système M1 – TP 9 : Sémaphores

Semaine du 23 Novembre 2009

Exercice 1 – Comptons

1. Écrire une fonction qui ouvre un fichier `toto` à l'aide de la primitive `open`, puis se divise en deux processus. Le père écrira dans `toto` tous les nombres pairs dans l'ordre, et le fils, tous les nombres impairs dans l'ordre.
2. Le fichier `toto` doit maintenant contenir tous les nombres entiers positifs *dans l'ordre croissant*. On synchronisera le père et le fils au moyen de deux sémaphores.
3. On veut maintenant que le père écrive tous les nombres pairs entre 0 et 50 et que le fils écrive tous les nombres impairs entre 0 et 100, le tout restant classé. `toto` contiendra donc :
...484950515355... Modifiez votre programme pour qu'il fonctionne dans cette situation.
4. Même question lorsque le père doit écrire plus de nombres que le fils.

On n'oubliera pas que le père doit attendre ses enfants, sous peine de générer des zombies.

Exercice 2 – Mémoire partagée

Refaire l'exercice 1 en utilisant un segment de mémoire partagée à la place du fichier `toto`. Le père affichera le contenu du segment après la fin du fils.

Exercice 3 – Des lecteurs et des écrivains

On dispose d'un fichier `tata`, 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. On la testera en générant un certain nombre de processus lecteurs et un certain nombre de processus écrivains, qui liront et écriront de manière aléatoire dans le temps.
 2. On veut autoriser un nombre maximal de processus lecteurs en même temps. Comment pouvez-vous réaliser ceci ?