

TP5 – Mémoire partagée et sémaphores (2)

Rappel

- 👉 Le TP sera noté. La copie directe des fichiers (maquillée ou non) sera sanctionnée pour les 2 (copieur et copié).
- 👉 Pour les sémaphores, vous utiliserez votre fichier `mes_semaphores.c` du dernier TP : il faut que les fonctions P et V fonctionnent.

Le TP est à réaliser progressivement dans l'ordre partie par partie et ► par ►.

Classement coopératif

Soit un tableau de $2N$ entiers. On souhaite regrouper dans la première moitié, les N plus petits éléments et dans l'autre moitié les N plus grands.

Exemple : avec le tableau

12	3	56	5	32	7
----	---	----	---	----	---

 on souhaite placer 3 7 et 5 dans la première moitié du tableau (et donc 12 56 et 32 dans la deuxième).

On propose pour résoudre ce problème en utilisant deux processus : *Père* et *Fils*.

- *Père* va rechercher le plus **grand** entier de la première partie du tableau pour le donner à *Fils*
- *Fils* recherchera le plus **petit** de la deuxième partie pour le donner à *Père*.

L'idée est d'échanger ces deux entiers jusqu'à l'équilibre du système. On ne cherchera pas classer les entiers dans chacune des parties.

Un seul des deux processus réalisera l'échange. Par convention, on considérera qu'il s'agit ici de *Fils*. Et pour éviter une communication de la valeur de la position entre les deux processus, *Père* écrira **en première position du tableau** l'élément qu'il a trouvé.

Exemple : Ici, on obtient à la fin le tableau

7	3	5	56	32	12
---	---	---	----	----	----

Ainsi les étapes sont les suivantes :

<i>Père</i>	<i>Fils</i>
Recherche du + grand	Recherche du + petit
Échanger avec la position 1 le + grand	
Attente du Fils (synchro)	Attente du Père (synchro)
	Échanger avec la position 1 le + petit
Attente du Fils (synchro)	
Si changement, recommencer	Si changement, recommencer

Déroulement de l'exemple :

12	3	56	5	32	7
56	3	12	5	32	7
5	3	12	56	32	7
12	3	5	56	32	7
7	3	5	56	32	12

Le père et le fils seront codés dans le même fichier (= pas d'utilisation de *excl*).

- a. **SEGMENT DE MÉMOIRE PARTAGÉE** : Le tableau à trier est partagé, il semble donc logique d'utiliser un segment de mémoire partagée pour le stocker.
- ▶ Réalisez le code pour le créer, le remplir et le détruire. Créez aussi une fonction d'affichage du tableau aussi qu'une fonction qui recherche le minimum d'un tableau entre 2 cases. Exemple : `minimum(tab, 3, 12)` recherche la valeur minimum dans les cases de 3 à 12 et renvoie la position dans le tableau `tab`.
 - ▶ Faites aussi la fonction qui recherche le maximum.
 - ▶ Faites en sorte que le Fils cherche 1 fois le minimum (de sa partie) et l'affiche, et que le Père cherche une fois le maximum (de sa partie) et l'affiche.
 - ▶ (le Père attendra la mort du Fils avant de détruire le segment)
 - ▶ Pour l'initialisation des valeurs du tableau, on utilisera des valeurs aléatoires. N (la taille du demi tableau) pourra être donné en argument du programme.



Nombre aléatoire

✎ Pour créer un générateur aléatoire de nombre :

```
/* Initialisation du générateur */
srand(time(NULL)) ou srand(getpid());
/* génération d'un nombre */
int i = rand(); /* nombre entre 0 et RAND_MAX (exclus) */
float f = rand()/(float)RAND_MAX; /* ici entre 0 et 1. (exclus) */
int j = (rand()/(float)RAND_MAX)*N; /*ici entre 0 et N (exclus) */
```

- b. **SYNCHRONISATION** : Pour synchroniser les processus avant de faire l'échange, on souhaite réaliser un mécanisme de rendez-vous entre les deux processus. Pour cela, on utilisera des sémaphores.
- ▶ **Avant de coder** : En reprenant le tableau des étapes, indiquez (en haut du fichier .c) les sémaphores que vous allez utiliser pour réaliser la le rendez-vous (pour l'instant sans l'échange), quelles sont leurs valeurs d'initialisation et à quel endroit vous placerez les `P()` et `V()`. [Rappel : pour faire un Rendez-vous avec deux sémaphores, on utilise des sémaphores privés... le nombre initial de jetons est de ...?]

- ▶ Réalisez cette synchronisation en utilisant vos fonctions P() et V(). Pour tester le rendez-vous le père fera un sleep(2) et le fils sleep(5) et les deux processus doivent s'attendre avant d'écrire le résultat de leur première recherche (du point précédent).
- ▶ Vous n'oublierez pas de détruire les sémaphores (par le père).
- ▶ Pour préparer la suite, avant le rendez-vous faites placer la valeur trouvée par le père dans la première case du tableau.

c. **ÉCHANGE ET FIN DE CLASSEMENT :**

- ▶ Réalisez l'étape de l'échange (voir ci-dessus). Le père doit attendre la fin de l'échange (réalisé par le fils) avant de recommencer à rechercher le plus grand élément de sa partie de tableau. Pour gérer la reprise (et l'attente) du père, on utilisera un nouveau sémaphore ou en plaçant judicieusement les sémaphores du premier rendez-vous... **Avant de coder :** Complétez en haut du fichier .c pour indiquer le(s) sémaphore(s) que vous allez utiliser pour réaliser l'échange, quelles sont leur valeur d'initialisation (s'ils sont nouveaux) et à quel endroit vous placerez les P() et V().
- ▶ Pour débbugger et aussi voir comment se comporte le programme le fils affichera le tableau juste avant l'échange et juste après.
- ▶ Lorsque le classement est fini, le fils doit prévenir le père (pour ne pas qu'il recommence à rechercher). Sachant que lorsqu'un fils se termine, il envoie automatiquement à son père le signal SIGCHLD, on souhaite profiter de ce comportement pour gérer la fin du classement. Le père affichera une dernière fois le tableau.

Dépôt du TP

Le TP est à déposer en 2 temps :

1. à la fin de la séance.
2. avant la date indiquée sur ARCHE.

Réalisez un fichier ZIP avec les fichiers du TP (dont mes_sempahores.c) et n'oubliez les .h s'il y en a.).

Pour le dernier dépôt, respectez les consignes suivantes :

- 👉 Indiquez comment se compilent vos programmes : idéalement réalisez un makefile (si vous savez faire), sinon indiquez les commandes de compilation dans un README.
- 👉 ATTENTION : pour les clés n'utilisez pas de chemin absolu ou de fichiers particuliers ! qui ne fonctionneraient pas sur une machine différente.
- 👉 Dans le *main*, réalisez un affichage de l'usage lorsque l'on tape le nom de l'exécutable seul (= gérez le nombre d'arguments donnés).

Ces dernières consignes permettront d'alléger la note de quelques points si par malheur elles n'étaient pas respectées.