

# Travaux Dirigés Programmation Système: Feuille 7

## Informatique 2ème année. ENSEIRB 2018/2019

—*Mathieu Faverge* - *mfaforge@enseirb.fr* —

### Threads et section critique

#### ►Exercice 1. Vendeurs de tickets :

Écrire un programme qui lance 8 threads, où chaque thread joue le rôle d'un vendeur de tickets : à partir d'un nombre de tickets initial, partagé entre tous les threads et qu'on lira sur l'entrée standard (on utilisera `atoi` pour faire la conversion), chaque thread fera en boucle :

- Tester si le nombre de billets à vendre est positif
- S'il est positif :
  - Faire un `pthread_yield()`
  - Décrémenter de 1 le nombre de billets disponibles,
  - Incrémenter le nombre de billets vendus par ce thread.
- Sinon, afficher le nombre de billets vendus par ce thread.

Expérimentez avec un nombre croissant de billets à vendre. Que se passe-t-il ?

Où est la section critique de ce code ?

Mettez la section critique en exclusion mutuelle à l'aide de sémaphores. On utilisera `sem_open(2)`, `sem_wait(2)`, `sem_post(2)` et `sem_close(2)`. Attention à ne pas arriver à la situation où un seul vendeur vend tous les tickets ! Vérifier qu'il n'y a plus le problème observé précédemment.

#### ►Exercice 2. Lecteurs et écrivains

Écrire un programme qui lance deux threads, un lecteur et un écrivain, partageant un tableau A commun d'un million d'éléments.

L'écrivain parcourt le tableau et en incrémente tous les éléments, affiche un message puis fait `sleep(1)`. Il effectue cette action 10 fois puis termine normalement.

Le lecteur parcourt le tableau et calcule la somme de tous ses éléments et affiche un message. Si la somme n'est pas paire, il affiche un message, puis fait `sleep(1)`. Il effectue cette action 10 fois puis termine normalement.

Le lecteur et l'écrivain doivent lire/écrire intégralement le tableau sans être interrompu par l'autre.

1. Quelle est la section critique de ce code ?
2. Tester le code avec un lecteur et un écrivain puis mettre en place à l'aide de sémaphores une exclusion mutuelle sur la section critique.
3. On veut maintenant pouvoir lancer plus d'un thread lecteur. Plusieurs lecteurs peuvent accéder simultanément au tableau A. En revanche, lecteurs et écrivain sont en exclusion mutuelle. Modifiez votre programme pour que 4 lecteurs soient lancés dans 4 threads et qu'un écrivain soit lancé dans un thread.
4. On souhaiterait maintenant donner la priorité à l'écrivain : dès que l'écrivain attend l'entrée en section critique, les lecteurs en cours peuvent finir, mais les nouveaux ne peuvent pas lui passer devant : ils sont mis en attente, le temps que l'écrivain finisse sa mise à jour de A. Modifiez votre programme en ce sens, en utilisant (au moins) un autre sémaphore.

#### ►Exercice 3. Dîner des philosophes

8 philosophes s'assoient autour d'une table ronde pour dîner. Il y a entre chaque convive 1 baguette. Chaque philosophe a besoin de sa baguette de gauche et de celle de droite pour pouvoir commencer à dîner.

Écrire un programme qui lance 8 threads et qui simule ce banquet. L'utilisation de chaque baguette sera protégée par un mutex. La baguette étant disponible si le mutex n'est pas pris, et est utilisée par un autre philosophe si il est pris. On supposera également que les philosophes commencent à prendre

leur baguette de droite avant de prendre celle de gauche. Chaque thread ayant eu accès aux deux baguettes appellera `sleep(1)` avant de les reposer. Chaque philosophe prendra les baguettes, mangera puis les reposera un nombre de fois assez grand pour révéler d'éventuels problèmes entre threads.

Que se passe-t-il ?

Proposer une modification du programme pour éviter ce problème.