

SR02 : TD 5

TD Exercices sur une séance de 2h

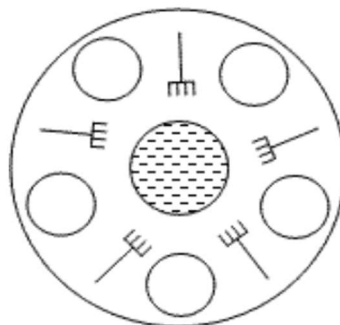
Objectifs

- Exploiter les sémaphores pour résoudre des problèmes de synchronisation

Exercice 1. (Problème des philosophes mangeurs de spaghettis)

Problème des Philosophes

Cinq philosophes dînent autour d'une table ronde. Sur la table, sont disposés 5 assiettes, 5 fourchettes et un plat de spaghettis (toujours plein)



Problème des philosophes

L'activité d'un philosophe est de penser, et de temps à autre, lorsqu'il a faim, il mange des spaghettis. Pour manger les spaghettis, chaque philosophe doit utiliser deux fourchettes. Si un de ses voisins a déjà pris une fourchette, un philosophe ne peut pas manger. Lorsqu'un philosophe pense, il n'utilise aucune fourchette. Le processus philosophe est décrit de la manière suivante :

Processus Philosophe

répéter

penser ; {sans fourchettes}

manger ; {avec 2 fourchettes}

jusqu'à faut ;

Solution à base de sémaphores :

Les fourchettes sont représentées par un tableau de sémaphores et chaque philosophe est désigné par un entier compris entre 0 et 4. Le philosophe peut accéder aux fourchettes gauche(i) et droite (i+1) modulo 5. Les déclarations sont :

var

fourchettes : tableau[0..4] de sémaphore ;

i : entier ;

- Ecrire la procédure philosophe (i), en supposant que le philosophe prend la fourchette gauche, puis droite, mange, repose la fourchette gauche puis droite.
- Montrer comment cette solution peut mener à un blocage.
- Construire une solution sans blocage en ajoutant un sémaphore table représentant le nombre de place disponible à la table.

Exercice 2. (Le problème des lecteurs/rédacteur)

Le problème des lecteurs/rédacteur est le suivant : il s'agit de partager un fichier (ou une base de données) entre des processus lecteurs et des processus rédacteur. La cohérence est assurée si l'accès en écriture est exclusif, par contre plusieurs lectures peuvent se faire en parallèle.

- En utilisant des sémaphores, écrire les processus lecteurs et rédacteurs dans le cas où les lectures sont plus prioritaires que les écritures dès la première lecture.
- En utilisant des sémaphores, écrire les processus lecteurs et rédacteurs dans le cas où les lectures et les écritures ont une priorité égale. (une même file d'attente)
- En utilisant des sémaphores, écrire les processus lecteurs et rédacteurs dans le cas où les lectures sont prioritaires sur les écritures, quel que soit le moment où elles sont demandées : une lecture n'est pas retardée par une écriture en attente, mais seulement par une écriture en cours.
- En utilisant des sémaphores, écrire les processus lecteurs et rédacteurs dans le cas où les écritures sont prioritaires sur les lectures dès la première écriture : lorsqu'une lecture est demandée, elle est mise en attente, jusqu'à ce qu'il n'y ait plus d'écriture en cours ou en attente.

Exercice 3. (Synchronisation avec sémaphores)

Soient les deux processus P1 et P2 suivants. Ils se partagent deux sémaphores, S1 et S2 initialisés à 0.

- Quelle synchronisation a-t-on imposée sur les exécutions des procédures A1, A2, B1 et B2 ?
- Écrire le code afin d'imposer la même synchronisation pour 3 processus P1, P2 et P3 en utilisant 3 sémaphores.
-

init(S1, 0), init (S2,0), init(S3, 0);

P1

procedure A1 ;

....

procedure B1 ;

P2

procedure A2 ;

....

procedure B2 ;

P3

procedure A3 ;

....

procedure B3 ;

- Écrire le code afin d'imposer la même synchronisation pour N processus en utilisant N sémaphores.
- On peut résoudre le problème pour N processus avec uniquement deux sémaphores (et un compteur). Donner le code des processus dans ce cas.

Exercice 4. (Producteur / Consommateur avec tampon borné)

Un processus producteur produit des objets (des entiers) qui sont consommés par un processus consommateur. Les objets sont consommés dans l'ordre de production. Pour que les deux processus puissent s'exécuter en parallèle, avec des rythmes différents pour la production et la consommation, ils partagent un tampon de taille N. L'accès au tampon constitue la section critique pour les deux processus.

- En utilisant un tampon circulaire de taille N géré par deux indices *i_occupé* (l'emplacement à consommer) et *i_libre* (l'emplacement où produire), proposer une solution avec des sémaphores.