

TD 5 de Systèmes d'exploitation

Synchronisation par sémaphores

Exercice 1 : La piscine (Latteux, 1980)

Un utilisateur d'une piscine (nageur) doit tout d'abord trouver une cabine et un panier libres, ranger ses vêtements dans le panier, libérer la cabine et enfin déposer le panier à la consigne. Lorsqu'il a fini de se baigner, il va rechercher son panier et attend une cabine libre pour se changer. Il y a **np** Paniers et **nc** Cabines (**nc < np**).

- a) Définir les points et les contraintes de synchronisation.
- b) Écrire le programme d'un **nageur i**, en utilisant les sémaphores et les variables d'état suivantes :
 - **ndp** : nombre des demandes de paniers non satisfaites,
 - **npo** : nombre de paniers occupés,
 - **ncl** : nombre de cabines libres.
- c) Existe-t-il une possibilité d'interblocage ?

Exercice 2 : Rendez-vous par rapport à aux identificateurs des processus

Un ensemble de n processus P_0, P_1, \dots, P_{n-1} réalisent une opération sur une variable commune A . On ne connaît pas l'ordre d'arrivée des processus.

- 1) Les opérations sur la variable A sont effectuées dans un ordre de précedence strict : Un **processus i** (i : identificateur du processus) ne peut effectuer son opération sur la variable A que si le **processus i-1** a déjà fini la sienne.
- 2) Un processus ne peut continuer son exécution que lorsque tous les autres processus auront terminé leur opération sur la variable A .
 - a) L'ensemble de processus est constitué de 4 processus P_0, P_1, P_2 et P_3 . En utilisant uniquement des sémaphores, écrire les programmes des 4 processus.
 - b) L'ensemble de processus est constitué de n processus P_0, P_1, \dots, P_{n-1} . Ecrire le programme d'un **processus i**, en utilisant les sémaphores.

Exercice 3 : Rendez-vous par rapport à l'ordre d'arrivée des processus

Un ensemble de n processus P_0, P_1, \dots, P_{n-1} réalisent des opérations sur 2 variables communes v_1 et v_2 .

- 1) Les opérations sur la variable v_1 sont effectuées selon l'ordre d'arrivée des processus.
 - 2) Un processus ne peut poursuivre son exécution que lorsque tous les autres processus auront terminé leur opération sur la variable v_1 .
 - 3) Cependant, les opérations sur la variable v_2 sont réalisées dans l'ordre inverse de la variable v_1 . Un processus d'ordre j ne peut effectuer son opération sur la variable v_1/v_2 que si le processus d'ordre $j-1/j+1$ a déjà fini la sienne.
- Ecrire le programme d'un processus i (i : identificateur du processus), en utilisant les sémaphores.

Exercice 4 : Producteurs - consommateurs

Pour fabriquer un objet X on a besoin de 2 pièces A et B . On dispose de 3 processus P_1, P_2 et P_3 .

- P_1 fabrique une pièce A et la dépose dans le bac B_1 ,
- P_2 fabrique une pièce B et la dépose dans le bac B_2 ,
- P_3 retire une pièce A du bac B_1 et une pièce B du bac B_2 et il assemble ces 2 pièces.

Chacun des bacs peut contenir au maximum n pièces.

- 1) En utilisant les sémaphores écrire les programmes des 3 processus :
 - a) Définir et initialiser les sémaphores,
 - b) Ecrire les programmes des 3 processus.
- 2) On ajoute un quatrième processus P_4 pour faire l'assemblage de pièces avec le processus P_3 . Ecrire le programme du processus P_4 et modifier, si nécessaire, le programme du processus P_3 .

Exercice 5 : Le modèle des lecteurs/rédacteurs (Courtois et al, 1971)

Considérons deux classes de processus appelés **Lecteurs** et **Rédacteurs**. Ces processus se partagent un fichier. Les lecteurs peuvent seulement consulter le fichier et les rédacteurs peuvent seulement écrire sur le fichier. Les processus de ces deux classes doivent respecter les contraintes suivantes :

- Plusieurs lecteurs peuvent lire simultanément le fichier.
- Un seul rédacteur à la fois peut écrire sur le fichier.
- Un lecteur et un rédacteur ne peuvent pas utiliser en même temps le fichier.

Ecrire le programme de chacune des classes de processus (Lecteurs et Rédacteurs), en utilisant la technique de synchronisation par sémaphore, dans les cas suivants :

a) **Priorité aux lecteurs ;**

b) **Priorité aux rédacteurs ;**

Exercice 6 : Producteur-Consommateurs

Un enseignant communique avec ses **N** étudiants via un tampon de **M** cases. L'enseignant dépose des exercices dans ce tampon (un exercice par case) et les étudiants les retirent. Chaque exercice déposé par l'enseignant doit être retiré par tous les étudiants. En plus, chaque étudiant doit retirer les exercices dans l'ordre dans lequel ils ont été déposés. Une case du tampon n'est libérée que lorsque tous les **N** étudiants ont retiré l'exercice qu'elle contient. Chaque étudiant travaille selon son propre rythme.

a) En utilisant les sémaphores, écrire le programme de l'enseignant et le programme d'un étudiant.

b) En utilisant un nombre de sémaphores **inférieur ou égal à (M+4)**, écrire le programme de l'enseignant et le programme d'un étudiant.

Indications : *Raisonnement en séries d'exercices.*

Une série est constituée de **M** exercices ; les **M** premiers exercices déposés constituent la série 1, les **M** suivants constituent la série 2, ... Le tampon peut contenir des exercices de deux séries différentes.

Exercice 7 : Allocateur de ressources

Considérons une ressource banalisée ; le nombre d'exemplaires étant limité à **nmax** unités. Des processus ont besoin pour s'exécuter de **x** exemplaires de la ressource ($0 < x \leq nmax$). Lorsqu'un processus demande **x** unités et que ce nombre **x** est supérieur au nombre de ressources disponibles, aucune allocation n'a lieu et le processus est mis en attente (bloqué). Il devra être réveillé lorsqu'au moins **x** unités de la ressource seront devenues disponibles à la suite d'une restitution de ressources. On désire satisfaire le plus possible de processus bloqués dans l'ordre croissant des demandes :

On dispose de 2 procédures :

a) **Insérer(file, élément, ordre)** : permet d'insérer l'élément « élément » dans la file « file », selon l'ordre « ordre » qui peut être : croissant, décroissant, FIFO.

b) **Retirer(file, élément)** : permet d'affecter le premier élément de la file « file » à « élément » et le supprimer de la file.

1) Il y a **N** processus, dans le système, numérotés de 0 à **N-1**.

a) Définir et initialiser toutes les variables et les structures de données.

b) Ecrire les programmes des procédures **allouer(paramètres)** et **libérer(paramètres)**.

2) On ne connaît pas le nombre de processus dans le système