

TRAVAUX DIRIGES
PROGRAMMATION SYSTEME
Série 2 : Synchronisation de Processus

Exercice1 : Soit P0 et P1 deux processus parallèles se partageant deux ressources R1 et R2. Les algorithmes de ces deux processus sont écrits comme suit :

Shared semaphore s1=1, s2=1;

| | |
|--|--|
| <pre>P0 () { While (TRUE) { P (s1) utiliser R1 P (s2) utiliser R1 et R2 V (s1) V (s2) } }</pre> | <pre>P1 () { While (TRUE) { P (s2) utiliser R2 P (s1) utiliser R1 et R2 V (s2) V (s1) } }</pre> |
|--|--|

- 1) à quelle situation anormale peut conduire l'exécution de ces deux processus ?
- 2) donner une solution à ce problème.

Exercice2 : Soit m imprimantes à allouer à n processus :

Résoudre le problème en utilisant les sémaphores dans le cas où un processus peut demander 1 ou 2 imprimantes :

- si un processus demande 2 imprimantes, la requête est satisfaite si 2 imprimantes sont libres.
- A la restitution de 2 imprimantes le processus demandant 2 imprimantes sera servi en priorité par rapport aux processus demandant 1 imprimante.

Exercice3: Résoudre le problème du producteur consommateur en utilisant les sémaphores et un registre de taille infini.

Exercice 4 : On considère trois processus P1, P2 et P3. Le processus P1 produit des messages qu'il dépose dans un tampon T1. P2 prélève les messages contenus dans T1, les traite puis dépose les résultats dans un tampon T2. P3 prélève les messages contenus dans T2 et les consomme.

- 1) Ecrire les algorithmes de P1, P2 et P3 de façon à garantir le non-interblocage en utilisant les sémaphores.
- 2) On considère maintenant que les 3 processus travaillent sur un même tampon T (au lieu de T1 et T2). Réétudier la question 1.

Exercice5 :

A) Soit l'ordre d'arrivée suivant des processus. Donner l'ordre d'accès à la section critique suivant les 4 variantes du modèle Lecteurs/Rédacteurs :

Lect1, Redact1, Lect2, Redact2, Lect3, Redact3

B) Supposons, maintenant, que la section d'entrée et de sortie est instantanée (sauf blocage) et que la lecture dure 10 ms alors que l'écriture dure 30ms. Donner l'ordre d'exécution, selon les 4 variantes, des processus suivants avec leur dates d'arrivées :

Lect1(0), Redact1(5), Lect2(9), Redact2(15), Lect3(19), Redact3(22).

C) Voici la solution du problème lecteur/rédacteur version sémaphores:

1) Déroulez la solution pour la séquence de processus donnée dans A et B. De quelle variante s'agit-il?

2) Expliquez l'utilité de chaque variable dans cette solution.

```
int nbr_lect=0, nbr_red=0 ;
shared semaphore mutex1=1, mutex2=1, mutex3=1, S1=1, S2=1 ;

Lecteur() {
    P(mutex3) ;
    P(S1) ;
    P(mutex1) ;
    nbr_lect ++ ;
    if( nbr_lect==1) P(S2) ;
    V(mutex1) ;
    V(S1) ;
    V(mutex3) ;
    Lire () ;
    P(mutex1) ;
    nbr_lect -- ;
    if( nbr_lect==0) V(S2) ;
    V(mutex1) ;
}

Redacteur() {
    P(mutex2) ;
    nbr_red ++ ;
    if( nbr_red==1) P(S1) ;
    V(mutex2) ;
    P(S2) ;
    Ecrire () ;
    V(S2) ;
    P(mutex2) ;
    nbr_red -- ;
    if( nbr_red==0) V(S1) ;
    V(mutex2) ;
}
```

Exercice8 : Problème du coiffeur endormi :

Un salon de coiffure possède N chaises d'attente plus une chaise pour le coiffé et un coiffeur. Les conditions de travail sont les suivantes :

- i) si un client arrive et qu'il n'y a pas de chaise de libre alors, il quitte le salon.
- ii) Si aucun client n'est présent alors le coiffeur dort.
- iii) Un client entre, s'assois s'il y a une chaise, attend si le coiffeur est occupé sinon il réveille le coiffeur.

Résoudre ce problème de synchronisation entre le processus coiffeur et les processus clients en utilisant des sémaphores.

-FIN-