

Examen semestriel

Module de « Systèmes d'exploitation2 »

Corrigé

Exercice 1 : (07 points)

Soit X, Y et Z des variables numériques. P1 et P2 sont deux processus dont les codes sont les suivants :

```
Processus P1
Debut
    X := X+1
Fin
```

```
Processus P2
Debut
    Si X > 50 Alors
        Y := Y +1
    sinon
        Y:= Y/2
    Finsi;
    Z := Z + Y
Fin
```

Question 1 : Expliquez quel problème peut-on avoir si les deux processus s'exécutent simultanément.

Réponse :

Si les deux processus s'exécutent simultanément, il y'a un problème d'accès concurrents à la variable partagée X. On peut avoir un résultat incohérent si le processus P1 tente de modifier cette variable alors que le processus P2 tente de son coté de la consulter.

(3 points)

Question 2 : Corrigez le problème en utilisant des sémaphores. Réécrivez le code des deux processus.

Réponse :

La solution doit assurer l'accès en exclusion mutuelle à la variable partagée X. On utilise un sémaphore mutex , initialisé à 1.

```
Processus P1
Debut
    Wait(mutex)
    X := X+1
    Signal(mutex)
Fin
```

```
Processus P2
Debut
    Wait (mutex)
    Si X > 50 Alors
        Signal(mutex)
        Y := Y +1
    Sinon
        Signal(mutex)
        Y:= Y/2
    Finsi;
    Z := Z + Y
Fin
```

(4 points)

Exercice 2 : (13 points)

On reprend le problème de synchronisation du Producteur-Consommateur avec buffer circulaire et de taille limitée N, mais on suppose qu'il y'a maintenant deux producteurs P1 et P2 et un seul consommateur C. Le producteur P2 a une particularité : il produit et dépose deux messages à la fois (si toutefois, il y'a des places libres dans le buffer).

Travail à faire : Proposez un schéma de synchronisation de ce problème. Ecrivez le code des 3 processus P1, P2 et C. Soignez vos déclarations.

Réponse :

Déclarations :

In : Entier (Init à 0) indice du dernier élément déposé dans le buffer

Out : Entier (Init à 0) indice de l'élément prêt à être prélevé par le Consommateur

Mutex : Sémaphore (init à 1) assurant l'accès en exclusion mutuelle à la variable partagée In.

Empty : Sémaphore (Init à N) permettant de bloquer les producteurs si le buffer est plein.

Full : sémaphore (Init à 0) permettant de bloquer le consommateur s'il n'ya aucune case pleine dans le buffer.

(3 points)

Processus Producteur1	Processus Producteur2	Processus Consommateur
Début Cycle Produire un message dans ZoneP Wait(Empty); Wait(mutex) Buffer[In] :=ZoneP ; In :=In+1 mod N Signal(mutex) Signal(Full) Fin Cycle Fin.	Début Cycle Produire un message dans ZoneP1 et ZoneP2 Wait(Empty); Wait(Empty); Wait(mutex) Buffer[In] :=ZoneP1 ; In :=In+1 mod N Buffer[In] :=ZoneP2 ; In :=In+1 mod N Signal(mutex) Signal(Full) Signal(full) Fin Cycle Fin.	Début Cycle Wait(Full) ZoneC :=Buffer[Out] ; Out :=Out+1 mod N ; Signal(Empty) ; Consommer le message de ZoneC Fin Cycle Fin.

(3 points)