

Partie 5: Interblocage

Exercice 1 :

Pour traiter une image de taille T unités, un système composé de N processeurs à mémoire partagée, crée N processus. Chaque processus s'exécute sur un processeur et se charge de traiter une partie de l'image. Pour faire son traitement, un processus a besoin d'une unité de mémoire par unité d'image qu'il a à traiter, mais demande de la mémoire unité par unité au fur et à mesure de ses besoins. Si une demande ne peut être satisfaite, le processus demandeur est mis en attente (attente active). La libération des unités de mémoire allouées à un processus aura lieu lorsqu'il aura fini le traitement de toute sa partie. La mémoire disponible pour répondre aux demandes d'allocation des différents processus est de taille M unités.

Question 1: Algorithme du banquier

Pour éviter les interblocages, le système utilise l'algorithme du banquier. Supposez que $N = 4$, $T = 16$, $M = 8$ et qu'à l'état courant les processus P_1 , P_2 , P_3 et P_4 ont respectivement traité 1 sur 3 unités, 1 sur 4 unités, 1 sur 4 unités et 3 sur 5 unités (par exemple, **P1 : 1 sur 3 unités** signifie que le processus P_1 est chargé de traiter 3 unités de l'image et a seulement traité 1 unité).

- 1) Est-il possible d'atteindre une situation d'interblocage, si $T \leq M$? Si $T > M$? Justifiez votre réponse.
- 2) Vérifiez, en utilisant l'algorithme du banquier, si l'état courant est certain (sûr ou sauf).
- 3) Le processus P_3 demande une unité de mémoire. Doit-on la lui accorder ? Attention, vous devez répondre à cette question en utilisant l'algorithme du banquier.

Question 2: Caractérisation de l'interblocage

Considérez un état courant où chaque processus P_i détient C_i unités et a encore besoin de R_i unités pour poursuivre et terminer le traitement de sa partie.

- 1) Exprimez en fonction de M , C_i et R_i , $i=1$ à N , la condition qui caractérise la situation d'interblocage (c'est à dire : aucune ressource n'est disponible et il y a encore des besoins).
- 2) Sachant que $T = \sum_{i=1,N} (C_i + R_i)$, montrez que si $T < N+M$ alors il n'y a pas d'interblocage à l'état courant.

Exercice 2 :

Considérez un ensemble de processus composés de plusieurs producteurs et d'un seul consommateur dont les codes sont les suivants :

<pre>semaphore plein =0, vide=N, mutex=1 ; char T[N] ; int ip=0 ;</pre>	
<pre>Producteur () { int i, M ; char ch[N]; Repeter { M=Lire(ch,N); M= min(N,M); Pour i=1 à M pas 1 faire P(vide) ; P(mutex) ; Deposer(ch,M,ip) ; //insérer ch dans T ip = (ip+M)%N ; V(mutex) ; Pour i=1 à M pas 1 faire V(plein) ; } tant que vrai ; }</pre>	<pre>Consommateur () { int ic =0 ; char c ; Repeter { P(plein) ; P(mutex) ; c= T[ic] ; V(mutex) ; V(vide) ; Traiter(c) ; ic++ ; } tant que vrai ; }</pre>

1. Peut-on avoir une situation d'interblocage ? (si oui, expliquez en décrivant clairement une situation d'interblocage).
2. Complétez le code précédent de manière à éviter les situations d'interblocage.

Exercice 3 :

Une firme bien connue concurrente de Apple décide de lancer un nouvel appareil qui permet d'écouter de la musique de format mp3 et d'en « stocker » plus de 80 Go, soit 40Go de plus que le fameux « iPod ». Cependant, avant de faire l'envoi sur le marché, elle décide de vous engager pour vérifier si leurs choix technologiques ont été judicieux pour ce type d'application.

Ce fameux appareil possède 64 Mo de mémoire principale et 80 Go de mémoire secondaire. Il peut en tout temps n'exécuter que trois processus (en plus de l'OS) sur son processeur embarqué. Le processus le plus important parmi ces trois est l'Afficheur chargé en mémoire dès le départ (désigné par A). De plus, un processus Transfert (nommé T) permet de transférer les données d'un ordinateur à sa mémoire secondaire à partir d'un port externe de type « USB ». Finalement, le dernier processus Son (appelé S) permet d'envoyer de la musique à partir des données de la mémoire secondaire vers un port externe d'écouteur. Il est important de savoir que l'écoute de la musique est la principale fonction de cet engin. L'affichage et le son doivent toujours être parfaits, tandis que le transfert de données n'est que secondaire.

Le problème majeur de cet appareil technologique est qu'il n'y a qu'une seule façon de sortir des données vers l'extérieur. Effectivement, pour acheminer du son par les écouteurs ou des données via le port « USB », il faut passer par une ressource d'accès externe. Cette ressource, ainsi que la mémoire principale et la mémoire secondaire sont les trois ressources qui sont partagées entre les trois processus. La dernière ressource du système est l'écran qui n'est accessible que par le processus A.

À un instant donné, le système est dans l'état suivant :

$$\begin{array}{ll}
 E = [1 & 1 & 1 & 1] & A = [0 & 0 & 1 & 0] \\
 Alloc = 1 & 0 & 0 & 0 & R = 1 & 0 & 0 & 0 \\
 & 0 & 1 & 0 & & 0 & 0 & 1 & 0 \\
 & 0 & 0 & 0 & & 1 & 0 & 0 & 0
 \end{array}$$

1- Vérifiez en utilisant l'algorithme du banquier, si l'état courant est sûr.

2- Peut-on éviter des situations d'interblocage ? Justifier votre réponse

Exercice 4 :

Soient trois processus concurrents qui utilisent en exclusion mutuelle 6 ressources différentes (de A à F). Ces trois processus exécutent respectivement les codes suivants :

```

Processus_1()
{
    while(1){
        prendre (&D);
        prendre (&E);
        prendre (&C);
        // Utilisation des
        ressources;
        liberer(&D);
        liberer(&E);
        liberer(&C);
    }
}

```

```

Processus_2()
{
    while(1){
        prendre (&C);
        prendre (&B);
        prendre (&F);
        // Utilisation des
        ressources;
        liberer(&F);
        liberer(&B);
        liberer(&C);
    }
}

```

```

Processus_3()
{
    while(1){
        prendre(&A);
        prendre (&B);
        prendre (&E);
    }
}

```

Ces processus concurrents peuvent-ils entrer en interblocage ? Expliquez à l'aide d'un graphe. Si oui, peut-on l'éviter ? Justifiez brièvement.

Exercice 5 :

Dans un système électronique de transfert de fonds, il existe des centaines de processus identiques qui fonctionnent comme suit : Chaque processus lit une ligne de données qui spécifie la quantité d'argent, le numéro de compte à créditer et le numéro de compte à débiter. Il verrouille ensuite l'un après l'autre les deux comptes, transfère l'argent puis libère les verrous.

1. **Peut-on avoir des situations d'interblocage ?** Justifiez votre réponse
2. Si vous répondez oui à la question précédente, **proposez** une solution qui permet de les éviter. **Attention** : ne pas libérer l'enregistrement d'un compte avant que le transfert ne soit terminé.
3. **Peut-on les prévenir ?** Si oui, **expliquez** brièvement comment ?