

## Examen 2020-2021

### Exercice n° 1 : Interblocage

[.../7 points]

NB : Au niveau de cet exercice, il faut raisonner sur le pire des cas (le cas le plus défavorable).

1. 5 processus se partagent 11 ressources identiques qu'ils demandent une à une. Chaque processus demande au maximum 3 ressources. Montrer qu'il ne peut y avoir interblocage.
2. On souhaite généraliser le résultat de l'exemple précédent : N processus se partagent M ressources qu'ils demandent une à une. Chaque processus demande au maximum T ressources qui ne doit évidemment pas dépasser les M ressources disponibles. Montrer que si la somme des demandes totales de tous les processus  $N \times T$  est plus petite que  $M + N$  alors il ne peut pas y avoir interblocage ( $N \times T < M + N$ ).
3. Soit 5 processus qui se partagent 6 ressources qu'ils demandent une à une et si chaque processus demande au plus 2 ressources, il ne peut y avoir interblocage selon le résultat de la question 2 ci-dessus. Supposons maintenant que ces 5 processus se partagent :
  - 6 ressources d'une classe A qu'ils demandent une à une sans dépasser le total  $T_a = 2$ .
  - 11 ressources d'une classe B qu'ils demandent une à une sans dépasser le total  $T_b = 3$ .Montrer que, contrairement au cas où il n'y a qu'une classe de ressources, la présence de deux classes de ressources peut conduire à un interblocage si chaque processus peut demander les ressources une à une dans une classe ou l'autre, sans dépasser les totaux  $T_a$  de A et  $T_b$  de B.
4. On dispose toujours de 5 processus. Montrer qu'avec 7 ressources de classe A et 14 ressources de classe B, il n'y a pas d'interblocage.
5. On souhaite généraliser le résultat de l'exemple précédent. Il faut noter que le résultat de la question précédente (4°) est aussi vrai avec 6 ressources de classe A et 15 ressources de classe B ou avec 8 ressources de classe A et 13 ressources de classe B; ou ... ou avec 10 ressources de classe A et 11 ressources de classe B.

On suppose qu'on dispose de :

- N processus qui se partagent les ressources de deux classes A et B.
- Na ressources de classes A qu'ils demandent une à une sans dépasser le total  $T_a$ .
- Nb ressources de classes B qu'ils demandent une à une sans dépasser le total  $T_b$ .

Donner la condition générale en fonction de N, Na, Nb, Ta et Tb pour n'est pas en avoir interblocage.

### Question1:

#### Données :

Nombre de processus (N) = 5

Nombre de ressources (M) = 11

Chaque processus demande au maximum (T) = 3 ressources

**Analyse :** Pour qu'il y ait interblocage, chaque processus doit être en attente de ressources détenues par un autre processus, formant un cycle d'attente circulaire.

**Étape 1 : Comprendre la demande totale maximale** Chaque processus peut demander jusqu'à 3 ressources, donc la demande totale maximale pour les 5 processus est :

$$N \times T = 5 \times 3 = 15$$

**Étape 2 : Condition pour éviter l'interblocage** Selon la théorie de l'interblocage, pour éviter l'interblocage, la condition suivante doit être remplie

$$N \times T < M + N$$

Appliquons cette condition avec les valeurs données :

$$5 \times 3 < 11 + 5$$

$$15 < 16$$

Cette condition est vérifiée, ce qui indique qu'il ne peut pas y avoir interblocage.

## Question 2:

Pour expliquer cette condition avec des exemples réels et montrer pourquoi elle garantit l'absence d'interblocage, voici une explication détaillée :

### Généralisation de l'exemple précédent

**Données :**

- Nombre de processus (N)
- Nombre de ressources (M)
- Demande maximale par processus (T)

**Condition pour éviter l'interblocage :**

$$N \times T < M + N$$

**Explication de la condition :**

**Demande totale maximale :** Chaque processus peut demander jusqu'à T ressources.  
Donc, la demande totale maximale pour les N processus est :

$$N \times T$$

**Ressources disponibles :** Il y a M ressources disponibles.

**Condition de sécurité :**

La condition  $N \times T < M + N$  assure que la somme des demandes de tous les processus est inférieure au nombre total de ressources disponibles plus le nombre de processus. Cela signifie qu'il y a toujours assez de ressources pour que les processus puissent obtenir les ressources nécessaires sans qu'un cycle d'attente circulaire (qui cause l'interblocage) se forme.

**Exemple réel 1 :**

**Supposons :**

$N = 4$  (4 processus)

$M = 10$  (10 ressources)

$T = 3$  (chaque processus demande au maximum 3 ressources)

**Vérifions la condition :**

$$4 \times 3 < 10 + 4$$

$$12 < 14$$

La condition est respectée. Voyons pourquoi cela évite l'interblocage :

- **Demande totale maximale :** Les 4 processus peuvent demander au total  $4 \times 3 = 12$  ressources.
- **Ressources disponibles :** Il y a 10 ressources disponibles.
- **Ressources "virtuelles" supplémentaires :** Ajoutons 4 processus pour obtenir un total de  $10 + 4 = 14$  ressources "virtuelles".

**Question 3 :**

### **Données :**

- Nombre de processus ( $N$ ) = 5
- Ressources de la classe A ( $M_a$ ) = 6, chaque processus demande au plus  $T_a = 2$
- Ressources de la classe B ( $M_b$ ) = 11, chaque processus demande au plus  $T_b = 3$

### **Analyse :**

#### **Situation avec une seule classe de ressources :**

Si les 5 processus se partagent les 6 ressources de la classe A et chaque processus demande au maximum 2 ressources :

- Nombre total de ressources demandées :  $N \times T_a = 5 \times 2 = 10$

La condition pour éviter l'interblocage est

$$N \times T < M + N : 5 \times 2 < 6 + 5 : 10 < 11$$

Cette condition est vérifiée, donc il n'y a pas d'interblocage avec une seule classe de ressources.

#### **Introduction de deux classes de ressources :**

Maintenant, les processus peuvent demander des ressources des deux classes A et B.

- Classe A : 6 ressources,  $T_a = 2$
- Classe B : 11 ressources,  $T_b = 3$

#### **Possibilité d'interblocage :**

Lorsqu'il y a deux classes de ressources, les processus peuvent se retrouver dans une situation où ils détiennent des ressources d'une classe et attendent des ressources d'une autre classe.

Considérons le pire des cas où les demandes se croisent entre les deux classes.

#### **Exemple concret :**

- **Processus P1 :**
  - Détient 1 ressource de la classe A
  - Attend 1 ressource de la classe B
- **Processus P2 :**
  - Détient 1 ressource de la classe B
  - Attend 1 ressource de la classe A
- **Processus P3 :**
  - Détient 1 ressource de la classe A

- Attend 1 ressource de la classe B
- **Processus P4 :**
  - Détient 1 ressource de la classe B
  - Attend 1 ressource de la classe A
- **Processus P5 :**
  - Détient 1 ressource de la classe A
  - Attend 1 ressource de la classe B

Dans ce scénario, chaque processus détient une ressource d'une classe et attend une ressource d'une autre classe, formant un cycle d'attente circulaire. Voici comment le cycle d'attente peut se former :

1. P1 attend une ressource de la classe B détenue par P2
2. P2 attend une ressource de la classe A détenue par P3
3. P3 attend une ressource de la classe B détenue par P4
4. P4 attend une ressource de la classe A détenue par P5
5. P5 attend une ressource de la classe B détenue par P1

### **Conclusion :**

La présence de deux classes de ressources introduit la possibilité d'un interblocage en croisant les demandes entre les deux classes. Contrairement au cas où il n'y a qu'une seule classe de ressources (où la condition  $N \times T < M + N$  empêche l'interblocage), la possibilité pour chaque processus de demander des ressources de deux classes différentes permet la formation de cycles d'attente circulaires, conduisant à un interblocage.

### **Illustration du cycle d'attente :**

**P1 : (A, attente B) -> P2 : (B, attente A) -> P3 : (A, attente B) -> P4 : (B, attente A) -> P5 : (A, attente B) -> P1**

Chaque processus se retrouve bloqué en attendant une ressource détenue par un autre, formant ainsi un cycle d'attente circulaire typique d'un interblocage.

### Question 4:

Pour éviter l'interblocage, nous vérifions que la somme des demandes des processus est inférieure à la somme des ressources disponibles plus le nombre de processus :

- Pour la classe A :  $5 \times 2 = 10$  demandes maximum, 7 ressources disponibles.
- Pour la classe B :  $5 \times 3 = 15$  demandes maximum, 14 ressources disponibles.

Cependant, la condition  $N \times T < M + N$  doit être vérifiée pour chaque classe :

Pour la classe A :  $5 \times 2 < 7 + 5$  soit  $10 < 12$ , ce qui est vrai.

Pour la classe B :  $5 \times 3 < 14 + 5$  soit  $15 < 19$ , ce qui est vrai.

Donc, il n'y a pas d'interblocage.

## Question 5:

### Généralisation pour éviter l'interblocage avec deux classes de ressources

Pour éviter l'interblocage dans un système où  $N$  processus se partagent des ressources de deux classes (A et B), nous devons nous assurer que les demandes totales des processus pour chaque classe de ressources respectent certaines conditions.

### Conditions de non-interblocage

Pour qu'il n'y ait pas d'interblocage avec :

- $N$  processus
- $N_a$  ressources de classe A
- $N_b$  ressources de classe B
- Chaque processus demandant au plus  $T_a$  ressources de la classe A
- Chaque processus demandant au plus  $T_b$  ressources de la classe B

Les conditions générales sont :

$$N \times T_a < N_a + N$$

$$N \times T_b < N_b + N$$

### Explication de la condition

Ces conditions assurent que la demande totale des processus pour chaque classe de ressources est inférieure au nombre de ressources disponibles plus le nombre de processus. Cela garantit qu'il y a toujours suffisamment de ressources pour qu'au moins un processus puisse obtenir toutes les ressources qu'il demande sans attendre, évitant ainsi la formation d'un cycle d'attente circulaire.

## Illustration avec des exemples concrets

Exemple 1 :

- $N = 5$  processus
- $N_a = 6$  ressources de classe A
- $N_b = 15$  ressources de classe B
- $T_a = 2$
- $T_b = 3$

Vérifions les conditions :

$$5 \times 2 < 6 + 5$$

$$10 < 11$$

Cette condition est vérifiée.

$$5 \times 3 < 15 + 5$$

$$15 < 20$$

Cette condition est également vérifiée.

Donc, il n'y a pas d'interblocage.

Exemple 2 :

- $N = 5$  processus
- $N_a = 8$  ressources de classe A
- $N_b = 13$  ressources de classe B
- $T_a = 2$
- $T_b = 3$

Vérifions les conditions :

$$5 \times 2 < 8 + 5$$

$$10 < 13$$

Cette condition est vérifiée.

$$5 \times 3 < 13 + 5$$

$$15 < 18$$

Cette condition est également vérifiée.

Donc, il n'y a pas d'interblocage.

### Exemple 3 :

- $N = 5$  processus
- $N_a = 10$  ressources de classe A
- $N_b = 11$  ressources de classe B
- $T_a = 2$
- $T_b = 3$

Vérifions les conditions :

$$5 \times 2 < 10 + 5$$

$$10 < 15$$

Cette condition est vérifiée.

$$5 \times 3 < 11 + 5$$

$$15 < 16$$

Cette condition est également vérifiée.

Donc, il n'y a pas d'interblocage.

## Conclusion

Pour éviter l'interblocage dans un système avec deux classes de ressources, il est crucial de vérifier que la demande totale de chaque classe de ressources par les processus est inférieure au nombre de ressources disponibles plus le nombre de processus. Les conditions  $N \times T_a < N_a + N$  et  $N \times T_b < N_b + N$  assurent qu'il n'y a pas de formation de cycle d'attente circulaire, et donc pas d'interblocage.

Ces conditions peuvent être généralisées pour tout nombre de classes de ressources en appliquant la même logique : pour chaque classe de ressources  $C_i$  avec  $N_i$  ressources disponibles et chaque processus pouvant demander jusqu'à  $T_i$  ressources de cette classe, la condition à vérifier est :

$$N \times T_i < N_i + N$$

En respectant ces conditions pour chaque classe de ressources, le système peut fonctionner sans interblocage.

