

Fiche de TD Interblocage 2

Exercice 1

On considère un système composé de 4 types de ressources :

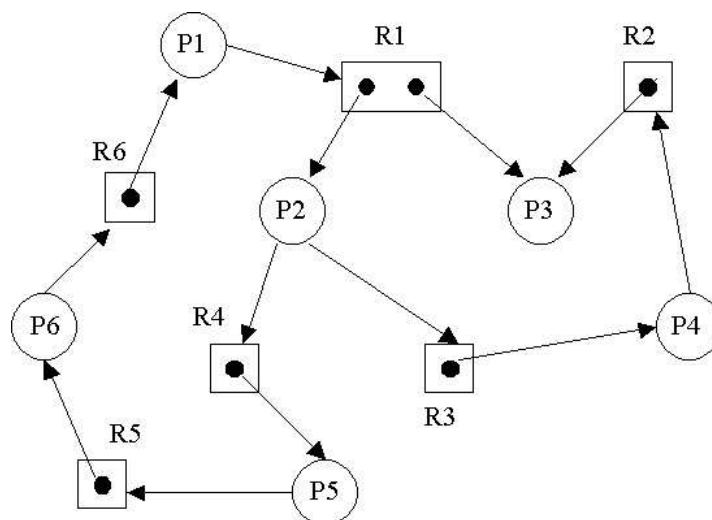
- Quatre (4) dérouleurs de bandes (DB)
- Deux (2) tables traçantes (TR)
- Trois (3) scanners (S)
- Un (1) lecteur de CD (CD)

Le système comporte trois (3) processus. Le processus P1 possède un scanner; le processus P2 possède deux dérouleurs de bandes et un lecteur CD; le processus P3 possède une table traçante et deux scanners. Chaque processus a besoin de ressources supplémentaires comme le montre la matrice Request :

	DB	TR	S	CD
P1	2	0	0	1
P2	1	0	1	0
P3	2	1	0	0

- Donner la matrice d'allocation
- Donner le vecteur de ressources disponibles
- Exécuter l'algorithme de détection d'inter-blocage pour vérifier si le système est en inter-blocage et si oui quels sont les processus impliqués
- Supposons que le processus 2 ait besoin de trois scanners et de la table traçante. Est-ce que le système est en état d'inter-blocage ?

Exercice 2



- Étant donné le graphe d'allocation de ressources de la figure ci-dessus. Choisissez la réponse correcte:
 - Le graphe a un cycle et donc on peut assurer qu'il n'y a pas d'interblocage.

- b) Le graphe a un cycle et donc on peut assurer qu'il y a un interblocage.
- c) Il y a un ordre d'ordonnancement des processus qui ne produit pas d'interblocage.
- d) Il y a un interblocage entre les processus P1, P2, P5 et P6.
- e) Aucune des réponses antérieures n'est correcte.

2. En utilisant le même graphe d'allocation des ressources, déterminer les vecteurs **E**, **Available** ainsi que les matrices **Allocation** et **Request** correspondantes aux processus P1, P2, P3, P4, P5 et P6.

3. Quel est l'état courant du système ?

Exercice 3

Considérons un système gérant quatre processus, P1 à P4, et trois types de ressources R1, R2 et R3 (3 R1, 2 R2 et 2 R3). Les ressources sont attribuées comme suit :

- P1 détient une ressource de type R1 et demande une ressource de type R2 ;
- P2 détient 2 ressources de type R2 et demande une ressource de type R1 et une ressource de type R3 ;
- P3 détient 1 ressource de type R1 et demande une ressource de type R2 ;
- P4 détient 2 ressources de type R3 et demande une ressource de type R1 ;

Y a-t-il un inter-blocage ? Si oui, quels sont les processus concernés?

Exercice 4

On considère quatre processus P1 P2 P3 P4 qui partagent un même fichier. Ce fichier est composé de 6 blocs de données notés B1, B2, B3, B4, B5 et B6. L'exécution de chacun des processus nécessite la pose de verrous exclusifs sur certains blocs :

P1 : B1 B3 B2 , P2 : B4 B2 B3, P3 : B5 B2 B1, P4 : B6 B2 B4 B3.

On Suppose qu'à l'état courant, les blocs suivants sont verrouillés :

B1 et B3 par **P1**, B4 par P2, B5 par **P3** B6 par **P4**

- a. L'état courant est-il sûr (sain, certain) ?
- b. Le système reçoit dans l'ordre les requêtes suivantes : P3 demande B2, P1 demande B2 et P2 demande B2. Le système répond favorablement à une requête uniquement si cela ne conduit pas vers un état non sûr. Indiquer, en utilisant l'algorithme du banquier, lequel des processus parviendra-t-il à verrouiller le bloc B2 ?

Exercice 5

L'état d'un système, à un instant donné, est représenté par les matrices suivantes de l'algorithme du Banquier :

Processus	Allocation				Max				Available			
	A	B	C	D	A	B	C	D	A	B	C	D
P0	0	0	1	2	0	0	1	2	1	5	2	0
P1	1	0	0	0	1	7	5	0				
P2	1	3	5	4	2	3	5	6				
P3	0	6	3	2	0	6	5	2				
P4	0	0	1	4	0	6	5	6				

- 1. Le système est-il dans un état sain ? . Justifiez.
- 2. Une requête (0, 4, 2, 0) arrive du processus P1. Peut-on l'accorder ? Justifiez.

Corrigé Type

Exercice 1 :

a. **Allocation** =

	DB	TR	S	CD
P1	0	0	1	0
P2	2	0	0	1
P3	0	1	2	0

$$= \begin{pmatrix} 0 & 0 & 1 & 0 \\ 2 & 0 & 0 & 1 \\ 0 & 1 & 2 & 0 \end{pmatrix}$$

b. **Available** = (E – Allocation) = (2 1 0 0)

a. Algorithme de détection d'interblocage : Request : $\begin{pmatrix} 2 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 2 & 1 & 0 & 0 \end{pmatrix}$

Work= (2 1 0 0)

Finish= (F, F, F)

Choisir i= 3 (Request₃ <= Work) → Work= (2, 2, 2, 0); Finish=(F, F, T)

Choisir i= 2 → Work= (4, 2, 2, 1); Finish=(F, T, T)

Choisir i= 1 → Work= (4, 2, 3, 1); Finish=(T, T, T)

Finish= (T, T, T) → **Pas d'interblocage.**

b. Request : $\begin{pmatrix} 2 & 0 & 0 & 1 \\ 0 & 1 & 3 & 0 \\ 2 & 1 & 0 & 0 \end{pmatrix}$

L'algo de détection d'interblocage :

Work= (2 1 0 0)

Finish= (F, F, F)

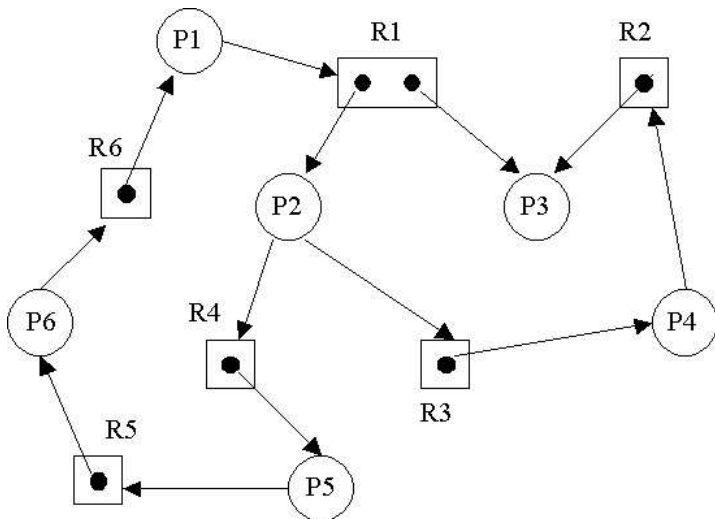
Choisir i= 3 → Work= (2, 2, 2, 0); Finish=(F, F, T)

∄ i tel que (Request₃ <= Work). Finish=(F, F, T) → **Interblocage.**

Exercice 2

1.

- c) Il y a un ordre d'ordonnancement des processus qui ne produit pas d'interblocage (Il y a un ordre de terminaison de processus, donc, il n'y a pas d'interblocage).



2.

$E = (2, 1, 1, 1, 1, 1)$

$Available = (0, 0, 0, 0, 0, 0)$

$$Allocation = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

$$Request = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

3. L'algorithme de détection d'interblocage :

$Work = (0, 0, 0, 0, 0, 0)$

$Finish = (F, F, F, F, F, F)$

Choisir $i = 3$ ($\text{Request}_3 \leq \text{Work}$) $\rightarrow \text{Work} = (1, 1, 0, 0, 0, 0)$; $\text{Finish} = (F, F, T, F, F, F)$

Choisir $i = 1$ ($\text{Request}_1 \leq \text{Work}$) $\rightarrow \text{Work} = (1, 1, 0, 0, 0, 1)$; $\text{Finish} = (T, F, T, F, F, F)$

Choisir $i = 4$ ($\text{Request}_4 \leq \text{Work}$) $\rightarrow \text{Work} = (1, 1, 1, 0, 0, 1)$; $\text{Finish} = (T, F, T, T, F, F)$

Choisir $i = 6$ ($\text{Request}_6 \leq \text{Work}$) $\rightarrow \text{Work} = (1, 1, 1, 0, 1, 1)$; $\text{Finish} = (T, F, T, T, F, T)$

Choisir $i = 5$ ($\text{Request}_5 \leq \text{Work}$) $\rightarrow \text{Work} = (1, 1, 1, 1, 1, 1)$; $\text{Finish} = (T, F, T, T, T, T)$

Choisir $i = 2$ ($\text{Request}_2 \leq \text{Work}$) $\rightarrow \text{Work} = (2, 1, 1, 1, 1, 1)$; $\text{Finish} = (T, T, T, T, T, T)$

$\text{Finish} = (T, T, T, T, T, T) \rightarrow \text{Pas d'interblocage.}$

Exercice 3

$\mathbf{E} = (3, 2, 2)$

$$\mathbf{Allocation} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 2 \end{pmatrix}$$

$\mathbf{Available} = (\mathbf{E} - \mathbf{Allocation}) = (1, 0, 0)$

$$\mathbf{Request} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

4. L'algorithme de détection d'interblocage :

$\text{Work} = (1, 0, 0)$

$\text{Finish} = (F, F, F)$

Choisir $i = 4$ ($\text{Request}_4 \leq \text{Work}$) $\rightarrow \text{Work} = (1, 0, 2)$; $\text{Finish} = (F, F, F, T)$

Choisir $i = 2$ ($\text{Request}_2 \leq \text{Work}$) $\rightarrow \text{Work} = (1, 2, 2)$; $\text{Finish} = (F, T, F, T)$

Choisir $i = 1$ ($\text{Request}_1 \leq \text{Work}$) $\rightarrow \text{Work} = (2, 2, 2)$; $\text{Finish} = (T, T, F, T)$

Choisir $i = 3$ ($\text{Request}_3 \leq \text{Work}$) $\rightarrow \text{Work} = (3, 2, 2)$; $\text{Finish} = (T, T, T, T)$

$\text{Finish} = (T, T, T, T) \rightarrow \text{Pas d'interblocage}$ (on a trouvé un ordre de terminaison des processus).

Exercice 4

a. Vérification de l'état sûr :

$$\mathbf{E} = (1, 1, 1, 1, 1, 1)$$

$$\mathbf{Allocation} = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\mathbf{Available} = (\mathbf{E} - \mathbf{Allocation}) = (0, 1, 0, 0, 0, 0)$$

$$\mathbf{Max} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 \end{pmatrix}$$

$$\mathbf{Need} = \mathbf{Max} - \mathbf{Allocation} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \end{pmatrix}$$

Vérification :

$$\mathbf{Work} = (0, 1, 0, 0, 0, 0)$$

$$\mathbf{Finish} = (\mathbf{F}, \mathbf{F}, \mathbf{F}, \mathbf{F})$$

Choisir $i = 1$ ($\text{Need}_1 \leq \text{Work}$) $\rightarrow \mathbf{Work} = (1, 1, 1, 0, 0, 0)$; $\mathbf{Finish} = (\mathbf{T}, \mathbf{F}, \mathbf{F}, \mathbf{F})$

Choisir $i = 2$ ($\text{Need}_2 \leq \text{Work}$) $\rightarrow \mathbf{Work} = (1, 1, 1, 1, 0, 0)$; $\mathbf{Finish} = (\mathbf{T}, \mathbf{T}, \mathbf{F}, \mathbf{F})$

Choisir $i = 3$ ($\text{Need}_3 \leq \text{Work}$) $\rightarrow \mathbf{Work} = (1, 1, 1, 1, 1, 0)$; $\mathbf{Finish} = (\mathbf{T}, \mathbf{T}, \mathbf{T}, \mathbf{F})$

Choisir $i = 4$ ($\text{Need}_4 \leq \text{Work}$) $\rightarrow \mathbf{Work} = (1, 1, 1, 1, 1, 1)$; $\mathbf{Finish} = (\mathbf{T}, \mathbf{T}, \mathbf{T}, \mathbf{T})$

$\mathbf{Finish} = (\mathbf{T}, \mathbf{T}, \mathbf{T}, \mathbf{T}) \rightarrow$ l'état actuel du système est un état **sûr** (sain).

b.

- P3 demande B2 : Exécuter l'algorithme du banquier.

Sauvegarder l'état du système (les matrices Available, Allocation et Need).

Allouer les ressources demandées par le processus P3 en modifiant l'état du système de la manière suivante :

$$\text{Available} := \text{Available} - \text{Request}_3 = (0, 0, 0, 0, 0, 0)$$

$$\text{Allocation} := \text{Allocation} + \text{Request}_3 = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\text{Need} := \text{Need} - \text{Request}_3 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \end{pmatrix}$$

Verification_Etat_Sain:

$$\text{Work} = (0, 0, 0, 0, 0, 0)$$

$$\text{Finish} = (F, F, F, F)$$

$\nexists i / \text{Need}_i \leq \text{Work}$ et $\text{Finish} = (F, F, F, F, F, F)$, donc l'allocation de B2 à P3 va mener à un **Etat non sain**.

La requête de P3 est rejetée → Restaurer l'ancien Etat du système.

- P1 demande B2 : Exécuter l'algorithme du banquier.

$$\text{Available} := \text{Available} - \text{Request}_1 = (0, 0, 0, 0, 0, 0)$$

$$\text{Allocation} := \text{Allocation} + \text{Request}_1 = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\text{Need} := \text{Need} - \text{Request}_1 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \end{pmatrix}$$

Verification_Etat_Sain:

$$\text{Work} = (0, 0, 0, 0, 0, 0)$$

$$\text{Finish} = (F, F, F, F)$$

Choisir $i = 1$ ($\text{Need}_1 \leq \text{Work}$) → $\text{Work} = (1, 1, 1, 0, 0, 0)$; $\text{Finish} = (T, F, F, F)$

Choisir $i = 2$ ($\text{Need}_2 \leq \text{Work}$) → $\text{Work} = (1, 1, 1, 1, 0, 0)$; $\text{Finish} = (T, T, F, F)$

Choisir $i = 3$ ($\text{Need}_3 \leq \text{Work}$) $\rightarrow \text{Work} = (1, 1, 1, 1, 1, 0)$; $\text{Finish} = (\text{T}, \text{T}, \text{T}, \text{F})$

Choisir $i = 4$ ($\text{Need}_4 \leq \text{Work}$) $\rightarrow \text{Work} = (1, 1, 1, 1, 1, 1)$; $\text{Finish} = (\text{T}, \text{T}, \text{T}, \text{T})$

$\text{Finish} = (\text{T}, \text{T}, \text{T}, \text{T}) \rightarrow \text{Etat sûr} \rightarrow \text{B2 est attribué à P1.}$

Exercice 5

1.

$\text{Available} = (1, 5, 2, 0)$

$$\text{Allocation} = \begin{pmatrix} 0 & 0 & 1 & 2 \\ 1 & 0 & 0 & 0 \\ 1 & 3 & 5 & 4 \\ 0 & 6 & 3 & 2 \\ 0 & 0 & 1 & 4 \end{pmatrix}$$

$$\text{Max} = \begin{pmatrix} 0 & 0 & 1 & 2 \\ 1 & 7 & 5 & 0 \\ 2 & 3 & 5 & 6 \\ 0 & 6 & 5 & 2 \\ 0 & 6 & 5 & 6 \end{pmatrix}$$

$$\text{Need} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 7 & 5 & 0 \\ 1 & 0 & 0 & 2 \\ 0 & 0 & 2 & 0 \\ 0 & 6 & 4 & 2 \end{pmatrix}$$

Vérification de l'état sain :

$\text{Work} = (1, 5, 2, 0)$

$\text{Finish} = (\text{F}, \text{F}, \text{F}, \text{F}, \text{F})$

Choisir $i = 0$ ($\text{Need}_0 \leq \text{Work}$) $\rightarrow \text{Work} = (1, 5, 3, 2)$; $\text{Finish} = (\text{T}, \text{F}, \text{F}, \text{F}, \text{F})$

Choisir $i = 2$ ($\text{Need}_2 \leq \text{Work}$) $\rightarrow \text{Work} = (2, 8, 8, 6)$; $\text{Finish} = (\text{T}, \text{F}, \text{T}, \text{F}, \text{F})$

Choisir $i = 1$ ($\text{Need}_1 \leq \text{Work}$) $\rightarrow \text{Work} = (3, 8, 8, 6)$; $\text{Finish} = (\text{T}, \text{T}, \text{T}, \text{F}, \text{F})$

Choisir $i = 3$ ($\text{Need}_3 \leq \text{Work}$) $\rightarrow \text{Work} = (3, 14, 11, 8)$; $\text{Finish} = (\text{T}, \text{T}, \text{T}, \text{T}, \text{F})$

Choisir $i = 4$ ($\text{Need}_4 \leq \text{Work}$) $\rightarrow \text{Work} = (3, 14, 12, 12)$; $\text{Finish} = (\text{T}, \text{T}, \text{T}, \text{T}, \text{T})$

Finish=(T, T, T, T, T) → **l'état est sûr.**

2. Une requête (0, 4, 2, 0) arrive du processus P1 : Algorithme du banquier.

Available := Available – Request₁ = (1, 1, 0, 0)

$$\text{Allocation} := \text{Allocation} + \text{Request}_1 = \begin{pmatrix} 0 & 0 & 1 & 2 \\ 1 & 4 & 2 & 0 \\ 1 & 3 & 5 & 4 \\ 0 & 6 & 3 & 2 \\ 0 & 0 & 1 & 4 \end{pmatrix}$$

$$\text{Need} := \text{Need} - \text{Request}_1 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 3 & 3 & 0 \\ 1 & 0 & 0 & 2 \\ 0 & 0 & 2 & 0 \\ 0 & 6 & 4 & 2 \end{pmatrix}$$

Vérification de l'état sain :

Work= (1, 1, 0, 0)

Finish= (F, F, F, F, F)

Choisir i = 0 (Need₀ <= Work) → Work= (1, 1, 1, 2); Finish=(T, F, F, F, F)

Choisir i = 2 (Need₂ <= Work) → Work= (2, 4, 6, 6); Finish=(T, F, T, F, F)

Choisir i = 1 (Need₁ <= Work) → Work= (3, 8, 8, 6); Finish=(T, T, T, F, F)

Choisir i = 3 (Need₃ <= Work) → Work= (3, 14, 11, 8); Finish=(T, T, T, T, F)

Choisir i = 4 (Need₄ <= Work) → Work= (3, 14, 12, 12); Finish=(T, T, T, T, T)

Finish=(T, T, T, T, T) → **Etat sain**

Donc la requête du processus P1 est accordée.