

Chapitre 6 - Interblocage

Table des matières

Chapitre 6 - Interblocage.....	1
1 - Introduction.....	3
2 - Modélisation des accès aux ressources.....	4
3 - Modélisation des interblocages.....	5
4 - Matrice de ressources.....	8
5 - Conclusion.....	10

1 - Introduction

Un système informatique dispose de nombreuses ressources à gérer et à allouer, qui ne peuvent l'être qu'un nombre limité de fois, souvent une seule.

Ainsi, il est courant que des tâches cherchent à obtenir l'accès à une ressource déjà allouée à une autre tâche, un problème se pose lorsque cette dernière attend également une autre ressource.

Ainsi, comme nous l'avons déjà vu, des situations d'interblocages sont possibles dans un système de tâches.

Andrew Tanenbaum définit l'interblocage de cette manière : « *Un ensemble de processus est en interblocage si chaque processus attend un événement que seul un autre processus de l'ensemble peut provoquer* ».

2 - Modélisation des accès aux ressources

Afin de détecter les interblocages, il est d'abord nécessaire de modéliser les tâches et les ressources.

Lorsqu'une tâche A (modélisée par un rond) demande une ressource R (caractérisée par un carré), ceci est représenté de la manière suivante :



Lorsqu'une ressource S (caractérisée par un carré) est attribuée à une tâche B (modélisée par un rond), ceci est représenté de la manière suivante :



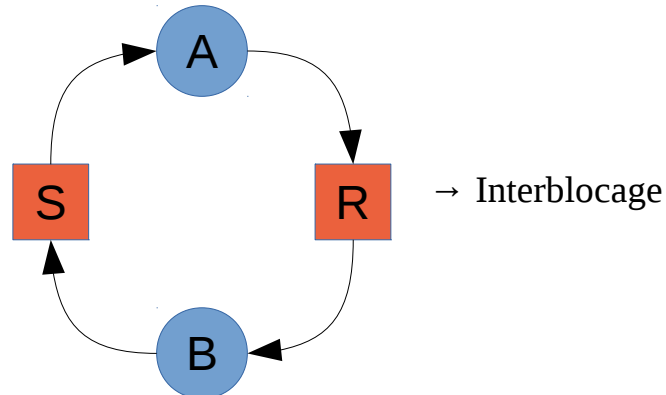
3 - Modélisation des interblocages

Pour modéliser les interblocages, il suffit de représenter un graphe des ressources allouées et attribuées, et de rechercher les cycles.

Considérons les contraintes suivantes :

- La tâche A détient la ressource S et demande la ressource R,
- La tâche B détient la ressource R et demande la ressource S,

Le graphe de ressources est donc le suivant :

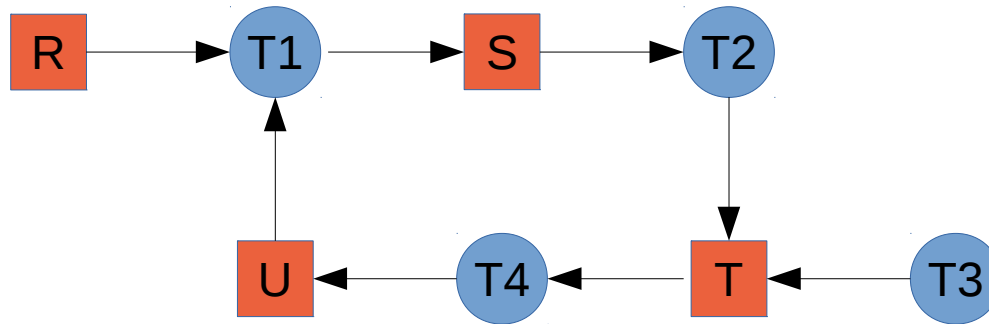


Le cas précédent est facile à détecter sans même représenter le graphe, mais regardons le cas suivant :

- La tâche T1 détient R et U et demande S,
- La tâche T2 détient S et demande T,
- La tâche T3 demande T
- La tâche T4 détient T et demande U

Il est facile visuellement de repérer un interblocage car il suffit de rechercher une boucle, cependant les algorithmes eux sont basés sur la recherche de cycle dans des graphes orientés.

Le graphe va faciliter la détection de l'interblocage :



4 - Matrice de ressources

Une autre solution lorsque de nombreuses ressources doivent être gérées est d'utiliser une modélisation avec des matrices et des vecteurs :

- E : Vecteurs de ressources totales de taille N ,
- A : Vecteurs de ressources disponibles de taille N ,
- C : Matrice des allocations de taille $M \times N$,
- R : Matrice des demandes de taille $M \times N$,

Avec :

- N le nombre de ressources à gérer,
- M le nombre de tâches,

L'idée est donc de passer de la matrice C_t à C_{t+1} , à partir d'une partie de la matrice R , en évitant les interblocages.

Exemple

- E : vecteur de ressources (4 2 1 3 6) représentant 5 ressources différentes,
- A_t : vecteur de ressources (1 1 0 1 2) représentant la disponibilité des 5 ressources différentes à l'instant t ,
- C_t : Matrice $\begin{pmatrix} 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 3 \\ 2 & 0 & 1 & 1 & 0 \end{pmatrix}$ des ressources allouées à l'instant t aux 3 tâches,
- R_t : Matrice $\begin{pmatrix} 0 & 1 & 0 & 0 & 2 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \end{pmatrix}$ des ressources demandés à l'instant t par les 3 tâches,

Ainsi, en analysant les différentes demandes et les ressource disponibles, il n'est possible de satisfaire les requêtes que pour les tâches T1 et T2.

5 - Conclusion

Comment font les systèmes d'exploitations actuels ?

- Les systèmes actuels réalisent en général la politique de l'autruche, c'est à dire que pour un bon nombre ressources, ils ne gèrent pas les interblocages car cela ralentirait l'ensemble du système.
- Statistiquement, cela a peu de chance d'arriver, et si c'est le cas, cela concernera des taches utilisateurs, donc une ou plusieurs applications, qui ne fonctionneront plus,