

# Interblocages (deadlocks)

## 1 Caractérisation

**Question.** Rappelez les quatre conditions qui peuvent conduire à un interblocage. Est-il nécessaire qu'elles soient toutes vérifiées pour qu'un interblocage puisse se produire ? Sont-elles indépendantes ?

**Question.** Rappelez la définition du graphe ressource-allocation.

Soit un système ayant 3 processus  $P_1, P_2, P_3$ , 4 types de ressources  $R_1, R_2, R_3, R_4$ , de nombre d'instances respectives 1, 2, 1, 3, se trouvant dans l'état suivant dans l'état suivant :

- $P_1$  détient une instance de type  $R_2$  et attend une instance de type  $R_1$  ;
- $P_2$  détient une instance de type  $R_1$  et de type  $R_2$  et attend une instance de type  $R_3$  ;
- $P_3$  détient une instance de type  $R_3$

**Question.** Quel est le graphe ressource-allocation associé ? Dessinez-le.

**Question.** Y a-t-il un risque d'interblocage ? Sur quelle propriété vous basez-vous ? Est-ce une condition nécessaire et suffisante ? Si oui, prouvez-le. Si non, donnez un contre-exemple et expliquez dans quelle cas cette propriété devient une condition nécessaire et suffisante. Que se passe-t-il si  $P_3$  demande une instance de type  $R_2$  ?

## 2 Prévention (prevention)

**Question.** Proposez/rappelez un moyen d'empêcher la réalisation de chacune des conditions pouvant amener à un interblocage. Jugez de la pertinence de ces propositions. Une attention particulière sera portée sur la *condition d'attente circulaire*. En quoi ces mécanismes des prévention sont-ils contraignants ?

## 3 Évitement (avoidance)

Un mécanisme d'évitement utilise de l'information supplémentaire pour éviter de rendre les interblocages possibles. Le modèle le plus simple et probablement le plus utile requiert que chaque processus déclare la quantité maximale de chaque type de ressources dont il peut avoir besoin. Nous nous basons sur ce modèle et définissons alors un état *sauf* comme il suit :

**Définition.** État sauf (safe) / aventureux (unsafe). *Un état est sauf si le système peut allouer les ressources à chaque processus (jusqu'à son maximum) dans un certain ordre et finalement éviter un interblocage. Un état qui n'est pas sauf sera qualifié d'état aventureux.*

**Question.** Quelles sont les relations entre états saufs, aventureux et d'interblocages ?

On suppose pour l’instant qu’il n’existe qu’une seule instance de chaque ressource.

**Question.** Proposez un algorithme d’évitement d’interblocage basé sur le graphe ressource-allocation. Vous pourrez introduire un nouveau type d’arête  $P_i \rightarrow R_j$  indiquant qu’un processus  $P_i$  peut demander  $R_j$  à un moment donné dans le futur. Illustrez votre algorithme sur un exemple comportant 2 processus et 2 ressources.

On suppose maintenant que plusieurs instances de chaque type de ressource peuvent exister. Soient  $n$  le nombre de processus et  $m$  le nombre de types de ressources. Soient les quatre structures de données suivantes :

**Dispo** Un vecteur de longueur  $m$  indiquant le nombre disponible de chaque type de ressource à l’instant présent.

**Max** Une matrices  $n * m$  indiquant la demande maximale de chaque processus sur chaque ressource (que chaque processus donne à l’entrée dans le système).

**Alloc** Une matrices  $n * m$  indiquant le nombre courant de chaque type de ressource alloué sur chaque processus.

**Besoin** Une matrices  $n * m$  indiquant la quantité d’instances de chaque type de ressource dont *peut* avoir de *surcroît* besoin chaque processus. On a :  $Besoin = Max - Alloc$ .

**Question.** Proposez un algorithme permettant de décider si un système est dans un état *sauf*. Quelle est sa complexité ?

**Question.** Déduisez-en un algorithme permettant de décider si une requête d’allocation peut-être satisfaite de manière *sauf*. Quelle analogie peut-on faire avec le problème d’un banquier proposant davantage de crédit qu’il ne possède de liquidité ?

## 4 Détection

Si aucun mécanisme de prévention ou d’évitement n’est utilisé, un interblocage peut avoir lieu. Dans ce cas, un système peut fournir un mécanisme baser sur la détection et le recouvrement. Nous nous intéressons ici à la partie détection.

On suppose pour l’instant qu’il n’existe qu’une seule instance de chaque ressource.

**Question.** Comment déterminer un interblocage à partir du graphe ressource-allocation. En déduire un mécanisme possible de détection. Quel est son coût ?

On suppose désormais que plusieurs instances de chaque type de ressource peuvent exister. Soient  $n$  le nombre de processus et  $m$  le nombre de types de ressources. Soient les quatre structures de données suivantes :

**Dispo** Un vecteur de longueur  $m$  indiquant le nombre disponible de chaque type de ressource à l’instant présent.

**Alloc** Une matrices  $n * m$  indiquant le nombre courant de chaque type de ressource alloué sur chaque processus.

**Requete** Une matrices  $n * m$  indiquant le nombre courant de requêtes supplémentaires de chaque processus sur chaque ressource.

**Question.** En vous inspirant de l’algorithme de la section précédente, proposez un algorithme de détection d’interblocage. Quelle est sa complexité.

## **A Rendons à César**

Ce sujet est tiré du cours *Operating System Concepts*, Silberschatz, Galvin, Gagne.