Travaux Dirigés n°2

Cours de Parallélisme

—Maîtrise Informatique—

Ce TD porte sur la prévention et la sortie d'un état d'interblocage. Nous abordons notamment la notion d'état sûr et voyons l'algorithme dit du banquier.

On représente un processus par un cercle et une ressource par un carré. On formalise alors la notion de demande de la ressource r_j par le processus p_i par le graphique de gauche ci-dessous, le graphique de droite signifiant au contraire que la ressource r_j est détenue par le processus p_i :

On définit par ailleurs les vecteurs d'informations suivants :

① **Dispo**$[j] = k$: lorsque que k ressources de type j sont disponibles ;

② **Max**$[i][j] = k$: lorsque le processus p_i peut demander au plus k instances de r_j ;

③ **Alloc**$[i][j] = k$: lorsque le processus p_i détient k instances de la ressource r_j ;

④ **Besoin**$[i][j] =$ **Max**$[i][j] -$ **Alloc**$[i][j]$.

Un état du système est caractérisé par le nombre de ressources disponibles, le nombre de ressources détenues par chaque processus et le nombre maximal de ressources nécessaires à chaque processus.

▸ **Exercice 1.** (notion d'état sûr) *Un état est qualifié de sûr dès lors qu'il existe un ordre total sur l'ensemble des processus $p_{o_0} < p_{o_1} < \cdots < p_{o_n}$ tel que toute demande de ressource par un processus p_{o_i} peut être satisfaite en utilisant les ressources disponibles ou celles actuellement détenues par un processus p_{o_j} tel que $o_j < o_i$. Un état est qualifié de peu sûr s'il ne vérifie pas les conditions d'un état sûr.*

1. *Donner un exemple utilisant le formalisme précédent montrant un ensemble de processus dans un état peu sûr.*
2. *On considère un ensemble de processus dans un état peu sûr, mais pas dans un état d'interblocage. Est-il possible de toujours terminer l'exécution de l'ensemble de ces processus ? Le cas échéant, on donnera un exemple montrant que partant d'un état peu sûr on peut arriver à un état d'interblocage.*
3. *Montrer que tout état sûr ne peut jamais correspondre à une situation d'interblocage.*
4. *En déduire une idée permettant à un système de ne jamais se trouver en situation d'interblocage.*

▸ **Exercice 2.** (algorithme du banquier) *On considère le vecteur* $\mathbf{R}[i]$ *de requètes du processus* p_i *de sorte que si* p_i *demande* k *instances de la ressource* r_j*, alors* $\mathbf{R}[i][j] = k$*. Cet algorithme s'organise autour de trois étapes qui sont les suivantes :*

① *Si* $\mathbf{R}[i] \leq \mathbf{Besoin}[i]$ *Faire* ② *Sinon erreur("demande trop importante") FinSi ;*

② *Si* $\mathbf{R}[i] \leq \mathbf{Dispo}[i]$ *Faire* ③ *Sinon attendre FinSi ;*

③ *Si le nouvel état est sûr Faire* $\mathbf{Alloc}[i] = \mathbf{Alloc}[i] + \mathbf{R}[i]$, $\mathbf{Dispo}[i] = \mathbf{Dispo}[i] - \mathbf{R}[i]$, $\mathbf{Besoin}[i] = \mathbf{Besoin}[i] - \mathbf{R}[i]$ *FinSi.*

On considère l'état d'un système de 4 *ressources partagées par* 5 *processus de la manière suivante au temps* t *:*

	Alloc	**Max**	**Dispo**
p_1	[0012]	[0012]	[1520]
p_2	[1000]	[1750]	[1520]
p_3	[1354]	[2356]	[1520]
p_4	[0632]	[0652]	[1520]
p_5	[0014]	[0656]	[1520]

1. *Que contient au temps* t *le vecteur* **Besoin** *?*
2. *Le système est-il dans un état sûr ?*
3. *La demande* $\mathbf{R}[2] = [0420]$ *peut-elle être satisfaite immédiatement ?*
4. *Dans l'algorithme précédent, il faut savoir tester si un état du système est sûr. Donner un algorithme réalisant ce test.*
5. *Donner la complexité de l'algorithme du banquier.*

▸ **Exercice 3.** (que choisir ?) *Plutôt que de se maintenir dans un état sûr afin de ne pas tomber dans un état d'interblocage, on peut choisir de détecter périodiquement si le système se trouve dans un état d'interblocage.*

1. *Proposer un algorithme de détection d'une situation d'interblocage.*
2. *Quelles sont les limitations liées à la détection d'interblocage par rapport aux autres méthodes évitant les situations d'interblocage.*

▸ **Exercice 4.** (supplémentaire ?)

1. *Réduire l'algorithme du banquier au cas d'une seule instance de chaque ressource.*
2. *Montrer par un exemple que l'algorithme du banquier sur plusieurs ressources ne peut être implémenté par l'application de l'algorithme à chacune des ressources.*