

Miniprojets - B (complément au sujet P4)

Septembre 2016

P4 : Système d'allocation de ressources (sans interblocage)

Conditions nécessaires pour l'interblocage (Habermann)

- CNS1 **Des ressources non partageables**
- CNS2 Les processus conservent les ressources déjà obtenues lorsqu'ils en attendent d'autres.
- CNS3 Les **demandes de ressources sont bloquantes** et il n'y a pas de préemption (ou réquisition).
- CNS4 Il existe, dans ce contexte, une **chaîne circulaire de processus telle que chacun réclame les ressources possédées par le suivant.**

Dans les applications (logicielles), on essaye d'éviter (ou de prévenir) l'interblocage.

Les conditions d'évitement de l'interblocage sont :

- **chaque processus fournit des informations sur son comportement futur.**
- **A chaque opération on vérifie que le danger d'interblocage n'existe pas**
- **Chaque processus annonce au départ son besoin maximum**
- On s'assure qu'à tout moment il existe au moins une solution pour s'en sortir.
- *L'algorithme du banquier* est un exemple.

Algorithme du banquier :

Le banquier dispose d'un capital (en plusieurs devises)

Le client demande des prêts.

La faillite de la banque intervient si un client ne peut rembourser son prêt de façon définitive.

Le banquier ne pourra pas honorer lui-même ses autres engagements et les clients ainsi traités ne pourront pas non plus lui rembourser ce qu'ils ont déjà emprunté.

Contrat entre banquier et client

Le client :

- s'engage à déclarer au départ son besoin maximum pour chacune des devises
- s'engage à ne jamais demander plus de ressources que dans sa déclaration initiale pour chacune des ressources
- s'engage à rendre tout ce qu'il a emprunté au bout d'un temps fini.

Le banquier :

- s'engage à prêter ce qu'on lui demande, pas toujours immédiatement, mais au bout d'un temps fini.

Dans l'algorithme du banquier, **on considère des états du système.**

- On définit la notion d'**états fiables.**
 - Une allocation de ressources devrait faire passer le système d'un état à un autre état.
 - **L'allocation n'a effectivement lieu que si elle fait passer le système d'un état fiable à un autre état fiable.**
- Un état d'allocation est **fiable** si à partir de cet état, il est possible de fonctionner sans danger de faillite de la banque dans les conditions les plus mauvaises où chaque client va demander le maximum prévu dans chacune des devises.

Principe du déroulement des allocations

On part d'un état réalisable E_0 .

On cherche un processus p_i qui, s'il était exécuté seul à partir de l'état E_0 en utilisant la totalité de son annonce, peut aller jusqu'à son terme et donc libérer toutes les ressources.

Un tel processus vérifie : $Annonce[i, *] - Alloc[i, *] \leq Dispo$

Si ce processus existe, il peut libérer la totalité de ses ressources et faire passer le système dans un nouvel état vérifiant :

$$Dispo_{E_1} = Dispo_{E_0} + Alloc[i, *] \quad (+demande -demande)$$

On recommence en cherchant un autre processus p_j pouvant faire passer le système dans l'état E_2 , ainsi de suite, ...

On obtient une suite de processus dite **suite fiable**.

Un état E_k du système est dit fiable si à partir de cet état, il est possible de construire une suite fiable complète, c'est à dire une suite fiable contenant tous les processus.

Notations algorithmiques (pour inspiration)

On utilise des vecteurs et des tableaux (matrices) pour déclarer les ressources, l'état d'allocation des ressources.

Soit T un tableau, $T[i, *]$ désigne la ligne i du tableau

Soit un système avec N processus et M classes de ressources.

Chaque classe de ressource existe en plusieurs exemplaires.

→ $[nbr_1, nbr_2, \dots, nbr_n]$

Chaque processus a x exemplaires d'une ressource (x_1, \dots, x_m) .

$p \backslash r$	r_1	r_2	\dots	r_m
p_1				
p_2				
\dots				
p_n				

L'état du système est représenté par :

Ressource, Dispo : Vecteur $[0, M-1]$ of integer ;

les ressources maximales de chaque classe, les ressources couramment disponibles à un moment.

Annonce, Alloc : Tableau $[0..N-1, 0..M-1]$ of integer ;

les annonces maximales de besoin de ressources, les ressources déjà allouées.

$$Dispo[r] = Ressource[r] - \sum_{p=0}^{N-1} Alloc[p, r]$$

L'algorithme d'allocation doit garantir les assertions suivantes :

$\forall p : processus; 0 \leq Alloc[p, *] \leq Annonce[p, *] \leq Ressource$

$Dispo \geq 0$

Avec les notations spécifiques :

0 vecteur nul,

V_1, V_2 deux vecteurs distincts de taille $dimV$,

$V_1 \leq V_2$ ssi $V_1[j] \leq V_2[j] \quad \forall j = 0..dimV - 1$

Tous les états qui vérifient ces assertions sont réalisables.