Faire en classe inversée : lire ce doc à la maison puis faire les exos en classe → a-t-on compris ?

1 Problématique

Dans un système multi-processus l'ordonnanceur alloue le processeur à chaque processus selon une politique définie. Les processus s'entrelacent mais ont un espace d'adressage indépendant. Cependant ils peuvent avoir besoin d'accéder aux mêmes ressources (logicielles ou matérielles) et cette situation peut devenir bloquante.

Comment assurer le bon fonctionnement d'un système multi-processus ?

2 Ressource

Une ressource désigne toute entité dont a besoin un processus pour s'exécuter. La ressource peut être matérielle comme le processeur, un périphérique ou elle peut être logicielle comme une variable. Une ressource est caractérisée [1]

— Par un état : elle est libre ou occupée

— Par son nombre de points d'accès, c'est-à-dire le nombre de processus pouvant l'utiliser en même temps.

Nous pouvons par exemple évoquer le cas de la carte son. Ce périphérique nécessite un accès exclusif pour accéder au micro. Un seul logiciel pourra l'utiliser pour enregistrer un son.

3 Interblocage

3.1 Une situation

Considérons deux processus A et B et deux ressources R1 et R2 qui n'ont qu'un point d'accès :

— A doit s'allouer R1 et R2 avant de s'exécuter.

— B doit s'allouer R2 et R1 avant de s'exécuter.

FIGURE 1 – Deux processus et deux ressources

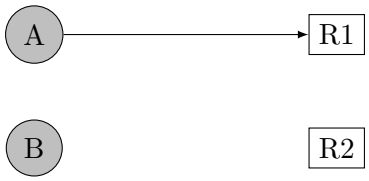

FIGURE 2 – L'ordonnanceur exécute A.
A demande R1 qui lui est allouée.

1. d'après le site http://y.legouzouguec.free.fr

Christophe Viroulaud

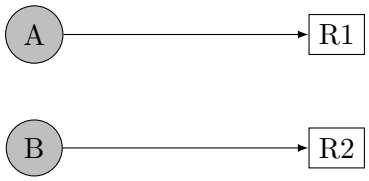

FIGURE 3 – L'ordonnanceur commute.
B demande R2 qui lui est allouée.

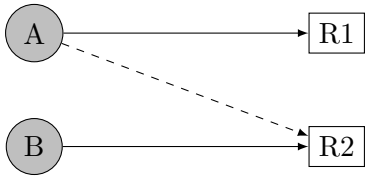

FIGURE 4 – L'ordonnanceur commute.
A demande R2 qui est déjà allouée. A est bloqué

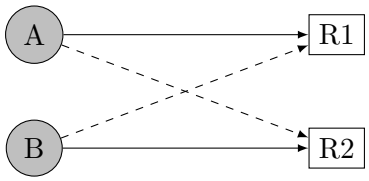

FIGURE 5 – L'ordonnanceur commute.
B demande R1 qui est déjà allouée. B est bloqué

Les processus A et B sont en situation d'interblocage.

3.2 Les conditions de l'interblocage

L'interblocage est le grand danger de la programmation concurrente. Il existe quatre conditions nécessaires à la présence d'un interblocage, décrites par Edward Grady Coffman en 1971 :

- **Exclusion mutuelle :** au moins une ressource doit être en accès exclusif.
- **Rétention et attente :** un processus détient une ressource et demande une autre ressource détenue par un autre processus.
- **Non préemption :** une ressource détenue par un processus ne peut être récupérée de force (*préemptée*) par un autre processus.
- **Attente circulaire :** chaque processus attend une ressource détenue par un des autres.

4 Traitement de l'interblocage

Sans rentrer dans le détail, nous pouvons citer les politiques mises en place pour traiter l'interblocage :

- **Politique de guérison :** le système maintient un état permanent des demandes de ressources et résout les éventuels interblocages en tuant un processus.
- **Politique de prévention :** le système fait en sorte que les quatre conditions ne soient jamais réunis. Par exemple, ne jamais donner une ressource si elle est déjà utilisée par un autre processus.
- **Politique de l'évitement :** à chaque demande de ressource, le système vérifie si cela peut causer un interblocage. Si tel est le cas, l'allocation est retardée.

— **Politique de l'autruche :** on ne s'en occupe pas, et on se contente de redémarrer la machine quand trop de processus sont en interblocage. Les trois autres politiques étant très coûteuses cette solution n'est finalement pas si farfelue.

exemples :

— accès base de données : deux processus A et B qui demandent des accès exclusifs aux enregistrements d'une base de données. A a verrouillé l'enregistrement R1 et demande l'accès à R2 et réciproquement.

— problème Edsger Dijkstra : 5 philosophes en cercle ; chacun un plat de spaghettis et une fourchette à gauche. Chaque philosophes a besoin de 2 fourchettes pour manger.

— circulation de voitures

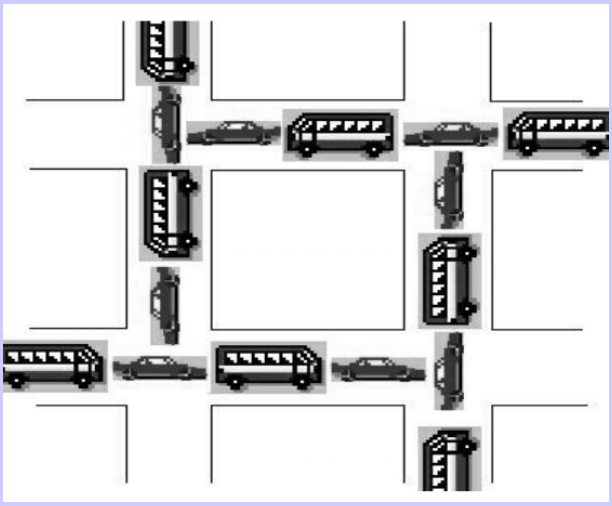

FIGURE 6 – priorité à droite

Christophe Viroulaud