

Travaux dirigés de Systèmes concurrents

φ

2 octobre 2023

Résumé

Ceci constitue les sujets des trois TD de systèmes concurrents :

1. TD moniteur : barbier, lecteurs/rédacteurs priorité aux rédacteurs
2. TD moniteur : lecteurs/rédacteurs fifo, allocateur de ressources fifo
3. TD processus communicants : voie unique, tournoi de bridge

1 Le problème du barbier

Il s'agit de modéliser l'activité coordonnée d'un barbier et de ses clients, à l'aide de processus synchronisés.

La boutique du barbier comporte un salon avec un unique fauteuil, et une salle d'attente avec N chaises. Lorsqu'un client se présente, il entre dans la salle d'attente s'il y a des chaises libres, sinon il attend dehors. Chaque fois que le barbier a fini de raser un client, il le lui signale, ce dernier quitte la boutique, puis le client suivant peut entrer dans le salon, pour se faire raser. Les clients entrent dans le salon un par un, lorsque vient leur tour. Lorsque le barbier n'a plus de client à raser, il s'endort. Lorsqu'un client trouve le barbier endormi et que c'est son tour, il le réveille et se fait raser.

Questions à étudier

- Donner une solution au problème avec un moniteur ;
- Considérer le cas où un client abandonne si la salle d'attente est pleine ;
- Considérer le cas où il y a plusieurs barbiers et fauteuils dans le salon.

2 Les lecteurs et les rédacteurs

On envisage deux classes de processus, des lecteurs et des rédacteurs, accédant à un fichier partagé. Ce fichier peut être lu simultanément par plusieurs lecteurs mais un seul rédacteur peut écrire à un instant donné, et à condition qu'aucun lecteur ne lise (exclusion mutuelle entre lecteurs et rédacteurs, et exclusion mutuelle entre rédacteurs).

Les actions assurant le protocole d'accès au fichier partagé sont les suivantes :

Débuter_Lecture,
Terminer_Lecture,
Débuter_Écriture,
Terminer_Écriture

Bon comportement : une activité qui a demandé la lecture (resp. l'écriture) finit par la terminer, et une activité en cours de lecture ou d'écriture ne redemande pas un accès. Une activité peut être modélisée par l'expression régulière : $((DL;TL) + (DE;TE))^*$.

Questions On étudiera les stratégies suivantes, réalisées avec un moniteur :

- Les rédacteurs sont prioritaires sur les lecteurs. Un lecteur ne peut accéder au fichier que si un rédacteur est en attente, même s'il y a déjà des lecteurs en cours. Un lecteur ne peut cependant pas être interrompu (expulsé) par un rédacteur.
- FIFO : on respecte l'ordre d'arrivée. Pour autant, on souhaite garder le parallélisme des lecteurs arrivés "ensemble". Par exemple, si on observe des arrivées L,L,L,R,L,L, les trois premiers lecteurs doivent passer, le rédacteur et les deux lecteurs suivants doivent attendre.
- En TP, les étudiants étudieront une stratégie équitable (absence de famine que ce soient des lecteurs ou des rédacteurs).

3 Allocateur de ressources critiques

Un système comporte généralement des ressources critiques c'est-à-dire non partageables et non préemptibles, comme les pages mémoire. Lorsqu'un processus utilise une page mémoire, aucun autre processus ne peut et ne doit y accéder. De plus, un processus ne doit pas attendre la libération d'une page mémoire occupée s'il existe par ailleurs des pages disponibles.

On se place dans le cas où il n'y a qu'une seule catégorie de ressources et on cherche un protocole permettant à un processus d'acquérir plusieurs ressources en une seule action. On ne se préoccupe pas de l'allocation effective des ressources proprement dites et on se concentre sur la coordination entre demandeurs et libérateurs de ressources. On considère donc que les actions assurant le contrôle d'accès à de telles ressources sont les suivantes :

```
Allouer(nbDemandé : entier);  
Libérer(nbRendu : entier);
```

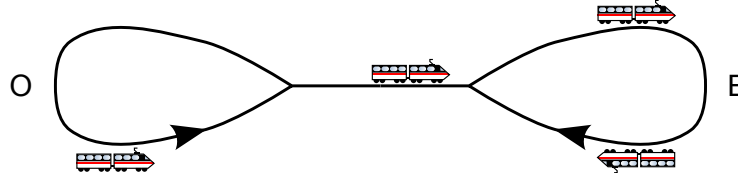
Bon comportement : un processus finit par rendre exactement le nombre de ressources obtenues, et un processus ne demande pas de ressources alors qu'il en possède déjà (risque d'interblocage complexe à gérer).

Questions

- En TD, on étudiera la stratégie FIFO, réalisée avec un moniteur.
- En TP, les étudiants considéreront la stratégie priorité aux petits demandeurs.

4 Voie unique

Deux villes E et O sont reliées par une ligne de chemin de fer qui comprend un tronçon à voie unique : ¹



Pour s'engager sur le tronçon à voie unique les processus représentant les trains devront faire appel aux actions suivantes, où *direction* appartient à $\{OE, EO\}$:

Entrer(*direction*)
Sortir

Les trains font la navette entre E et O. Leur comportement peut donc être simulé par le type de processus suivant :

```
process train is
  maDirection : Direction;
begin
  maDirection ← EO ou OE
  loop
    arriver au tronçon;
    Entrer(maDirection);
    passer le tronçon;
    Sortir;
    arriver à destination;
    maDirection ← inverse(maDirection);
  end loop
end train
```

Questions Proposer une solution basée sur des processus communicants (CSP/Go) dans les cas suivants :

1. un seul train peut se trouver sur le tronçon à voie unique ;
2. un nombre illimité de trains peuvent circuler en même temps sur le tronçon à voie unique (à condition qu'ils aillent dans le même sens) ;
3. au plus N trains peuvent circuler en même temps sur le tronçon à voie unique.

1. (train : source : wikicommons)

5 Tournoi de bridge

On désire modéliser le déroulement d'un tournoi de bridge (ou de belote). L'organisation du tournoi impose d'assurer que le nombre de personnes présentes dans la salle du tournoi soit toujours un multiple de 4. Il faut donc contrôler les entrées et les sorties selon les contraintes suivantes :

- un joueur peut entrer si un autre est prêt à sortir (échange) ou s'il y a déjà trois autres joueurs en attente d'entrée (le groupe de quatre peut alors entrer) ;
- un joueur peut sortir si un autre est prêt à entrer (échange) ou s'il y a déjà trois autres joueurs en attente de sortie (le groupe de quatre peut alors sortir).

Remarque : on ne s'intéresse pas à l'affectation des joueurs aux tables. Dans le cas de la sortie de quatre joueurs appartenant à des tables différentes, on ne s'intéresse pas à la réorganisation des tables de jeu pour continuer le tournoi.

Question Proposer une solution basée sur des processus communicants (CSP/Go).