

# Les sémaphores et mutex

## Objectifs du TP :

1. Manipuler les sémaphores et les mutex
2. Résoudre des problèmes à l'aide des sémaphores et des mutex.

## 1 Les sémaphores

Les sémaphores POSIX permettent aux processus et aux threads de se synchroniser. Un sémaphore est un entier dont la valeur ne peut jamais être négative. Deux opérations peuvent être effectuées : incrémenter la valeur du sémaphore de 1 (`sem_post(3)`), ou décrémenter la valeur du sémaphore de 1 (`sem_wait(3)`). Si la valeur courante est 0, une opération `sem_wait(3)` bloque jusqu'à ce que la valeur devienne strictement positive.

Deux utilisations sont faites des sémaphores :

- la protection d'une ressource partagée (par exemple l'accès à une variable, une structure de donnée, une imprimante...). On parle de sémaphore d'exclusion mutuelle ;
- la synchronisation de processus (un processus doit en attendre un autre pour continuer ou commencer son exécution).

Bien souvent on peut assimiler la valeur positive du compteur au nombre de processus pouvant acquérir librement la ressource.

**Question 1.1.** *Quelles sont les différences entre un sémaphore et un mutex ?*

## 2 Le salon de coiffure

Un coiffeur possède un salon avec un siège de coiffeur et une salle d'attente comportant un nombre fixe de fauteuils. S'il n'y a pas de client, le coiffeur se repose sur son siège de coiffeur. Si un client arrive et trouve le coiffeur endormi, il le réveille, s'assoit sur le siège de coiffeur et attend la fin de sa coupe de cheveux. Si le coiffeur est occupé lorsqu'un client arrive, le client s'assoit et attend sur une des chaises de la salle d'attente ; si la salle d'attente est pleine, le client s'en va. Lorsque le coiffeur a terminé une coupe de cheveux, il fait sortir son client courant et va réveiller un des clients de la salle d'attente. Si la salle d'attente est vide, il se rendort sur son siège jusqu'à ce qu'un nouveau client arrive.

**Question 2.1.** *Lister les comportements du coiffeur et des clients.*

**Question 2.2.** *Modéliser à l'aide de sémaphore et de mutex le comportement du coiffeur et de ces clients dans le salon de coiffure. Ecrire le programme C simulant l'activité du coiffeur.*

*Vous trouverez un canevas du programme : `/home/rbolze/teach/asr2/tp4-src/`.*

## 3 Le problème des philosophes.

Ce problème a été énoncé par Edsger Dijkstra. La situation est la suivante :

- cinq philosophes (initialement mais il peut y en avoir beaucoup plus) se trouvent autour d'une table ;
- chacun des philosophes a devant lui un plat de raviole ;
- à gauche de chaque assiette se trouve une fourchette.

Un philosophe n'a que trois états possibles :

- penser pendant un temps indéterminé ;
- être affamé (pendant un temps déterminé et fini sinon il y a famine) ;
- manger pendant un temps déterminé et fini.

Des contraintes extérieures s'imposent à cette situation :

- quand un philosophe a faim, il va se mettre dans l'état « affamé » et attendre que les fourchettes soient libres ;
- pour manger, un philosophe a besoin de deux fourchettes : les deux fourchettes qui entourent sa propre assiette ;
- si un philosophe n'arrive pas à s'emparer d'une fourchette, il reste affamé pendant un temps déterminé, en attendant de renouveler sa tentative.

Le problème consiste à trouver un ordonnancement des philosophes tel qu'ils puissent tous manger, chacun à leur tour.

**Question 3.1.** *Ecrire un programme permettant de modéliser le comportement des philosophes.*

**Question 3.2.** *Lister quelques cas problématiques.*

## 4 La salle de bain unisexe

Les douches des garçons du gymnase sont en travaux. Pendant ce temps, les garçons et les filles utiliseront les douches des filles. Afin de ne choquer personne, un panneau est accroché à l'entrée des douches : libre, occupé par des filles ou occupé par des garçons. Seules des filles peuvent entrer dans les douches lorsque des filles s'y trouvent déjà et réciproquement pour les garçons.

**Question 4.1.** *Créez un programme simulant un groupe de  $G$  garçons et un groupe de  $F$  filles attendant de prendre leur douche. Chaque personne sera simulée par un thread, et  $D$  cabines de douches seront disponibles.*

**Question 4.2.** *Existe-t-il une solution plus efficace que les autres ?*

**Question 4.3.** *Il est possible de donner la priorité à un groupe (les filles d'abord, par exemple), cette "stratégie" est-elle plus efficace qu'une solution équitable ? Mesurez le temps nécessaire pour que tout le monde se douche en fonction de la stratégie utilisée.*