

Problème_01 :

Programmer la gestion d'une station-service possédant N pompes à essence. Les voitures (modélisées par des threads) arrivent aléatoirement, et doivent :

- Attendre qu'une pompe soit libre
- S'en servir pendant un court instant (temps simulé)
- Libérer la pompe

Contraintes

- Une seule voiture par pompe à la fois
- Il peut y avoir plus de voitures que de pompes
- Pas de collision ni de dépassement de capacité

Problème_02 :

Source : https://en.wikipedia.org/wiki/Sleeping_barber_problem

Il s'agit ici d'un de ces problèmes de synchronisation mis sous une forme "plaisante".

Mais on en trouve une application presque directe dans certains mécanismes des systèmes d'exploitation (comme l'ordonnancement des accès disque).

La boutique du barbier est composée d'une salle d'attente contenant N chaises et d'un salon où se trouve la chaise du barbier. Lorsque le barbier a fini de raser un client, il fait entrer le client suivant dans le salon. Si la salle d'attente est vide, le barbier s'y installe pour dormir. Si un client trouve le barbier endormi, il le réveille. Si non, il s'installe dans la salle d'attente s'il reste de la place (ou rentre chez lui sinon).



Résoudre le problème avec les sémaphores.

Problème_03 :

Source : https://fr.wikipedia.org/wiki/D%C3%A9ner_des_philosophes

Cinq philosophes sont assis autour d'une table ronde. Entre chaque paire de philosophes se trouve une fourchette (soit 5 au total).

Pour manger, un philosophe a besoin des deux fourchettes adjacentes.

Chaque philosophe alterne entre :

- Réfléchir
- Tenter de manger (en prenant deux fourchettes)
- Manger
- Relâcher les fourchettes



Contraintes

- Chaque fourchette est une ressource partagée (sémaphore binaire).
- Il ne doit pas y avoir de blocage (deadlock).
- Il ne doit pas y avoir de famine (un philosophe qui ne mange jamais).