

## TD N° 3

### Exercice 1

On considère un ensemble de six tâches séquentielles {A, B, C, D, E, F}.

La tâche A doit précéder les tâches B, C, D. Les tâches B et C doivent précéder la tâche E. Les tâches D et E doivent précéder la tâche F.

Réaliser la synchronisation de ces tâches en utilisant les sémaphores.

### Exercice 2

Soient trois processus concurrents P1, P2 et P3 qui partagent les variables n et out. Pour contrôler les accès aux variables partagées, un programmeur propose les codes suivants :

Semaphore mutex1 = 1 ;

Semaphore mutex2 = 1 ;

Code Processus P1	Code Processus P2	Code Processus P3
P(mutex1) ; P(mutex2) ; out =out +1 ; n=n - 1 ; V(mutex2) ; V(mutex1) ;	P(mutex2) out =out - 1 ; V(mutex2) ;	P(mutex1) ; n=n + 1 ; V(mutex1) ;

- (a) Cette proposition est-elle correcte ? Si non, pourquoi ?  
 (b) Proposer une solution correcte.

### Exercice 3

Le problème à traiter concerne la mise au point d'un rendez-vous de travail entre N personnes. Chaque personne doit se rendre au lieu du rendez-vous et attendre le début de travaux qui ne peut commencer que lorsque tout le monde est présent. Le dernier arrivé donc débloquent la situation et les travaux de l'assemblée peuvent démarrer.

Résoudre ce problème en utilisant des sémaphores et considérant que les personnes sont des processus citoyens.

### Exercice 4

Deux villes A et B sont reliées par une seule voie de chemin de fer. Les trains peuvent circuler dans le même sens de A vers B ou de B vers A. Mais, ils ne peuvent pas circuler simultanément dans les sens opposés. On considère deux classes de processus: les trains allant de A vers B (Train AversB) et les trains allant de B vers A (Train BversA). Ces processus se décrivent comme suit :

Train AversB : Demande d'accès à la voie A; Circulation de A vers B; Sortie de la voie par B;	Train BversA : Demande d'accès à la voie B; Circulation de B vers A; Sortie de la voie par A;
--	--

Représenter les programmes des deux types de processus.

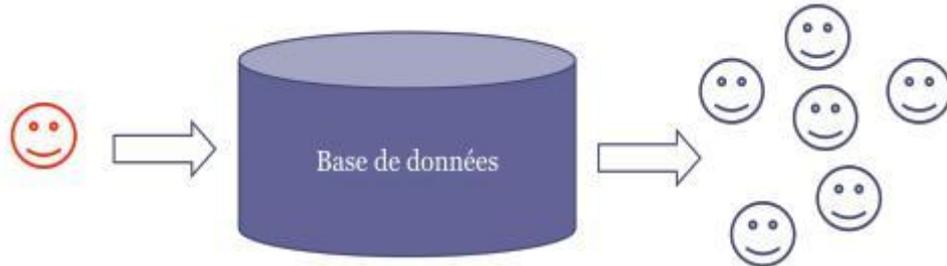
---

## Exercice 5

---

Le problème des lecteurs et des rédacteurs est un problème classique en théorie informatique, qui permet de modéliser les accès à des bases de données.

Soit un système de réservation de billets d'avion, par exemple, où de nombreux processus entrent en concurrence pour effectuer des lectures et des écritures. Il est acceptable que plusieurs processus puissent lire la base de données en même temps. Mais l'accès en écriture et les accès en lecture sont mutuellement exclusifs.



Ecrire le pseudo-code du processus lecteur et du processus rédacteur dans les deux cas suivant :

- version simplifiée : un seul lecteur est autorisé à se connecter à la base à la fois
- version généralisée : plusieurs lecteurs sont autorisés à se connecter en même temps.

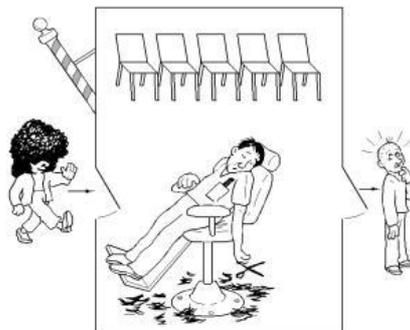
---

## Exercice 6

---

Il s'agit encore d'un de ces problèmes de synchronisation mis sous une forme "plaisante". Mais celui-ci est encore plus sérieux que le problème des philosophes, car on en trouve une application presque directe dans certains mécanismes des systèmes d'exploitation (comme l'ordonnancement des accès disque).

Un coiffeur possède un salon avec un siège de coiffeur et une salle d'attente comportant un nombre fixe  $F$  de fauteuils. S'il n'y a pas de clients, 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 s'endort sur une des chaises de la salle d'attente ; si la salle d'attente est pleine, le client repasse plus tard. 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.



Ecrire un pseudo-code permettant de synchroniser entre les deux processus : le coiffeur et le client.