TD Programmation Système

Série 6 : Opérations sur les sémaphores

Exercice 1 : Test des sémaphores

Soit P1 et P2 deux processus sans lien de parenté qui s'exécutent en parallèle. On vent programmer la synchronisation suivante:

-P2 est bloqué au départ

-P1 s'exécute, affiche la série : 2 4 6 8 10

et il se bloque

-P2 reprend son exécution est affiche la série : 3 6 9 12 15

En utilisant le module permettant la gestion des sémaphores (*semaphore.h, semaphore.c*) décrit en cours permettant la création, l'initialisation et les opérations sur les sémaphores (P et V) implémenter les programmes P1 et P2.

Indication :

Un sémaphore S permet de gérer la synchronisation entre P1 et P2. Le processus P2 utilise l'opération P(S) qui le met à l'état bloqué au début de son exécution. Le processus P1, affiche la série des multiples de 2 jusqu'au premier multiple de 5 et débloque le processus P2 par l'opération V(S).

Exercice 2 : Synchronisation par Sémaphores

Soit P1 et P2 deux processus qui partagent un segment de mémoire représentant un entier N et s'exécutant en parallèle. On désire avoir l'affichage suivant :

Terminal 1

```
P1 : 2 4 6 8 10
P1 : 27 29 31 33 35
P1 : 52 54 56 58 60
.....................

.....................
```

Terminal 2

```
P2 : 13 16 19 22 25
P2 : 38 41 44 47 50
P2 : 63 66 69 72 75
...................

...................
```

On s'arrête quand n > 100. Ecrire les deux programmes C sous Unix implémentant P1 et P2 en utilisant le module permettant la gestion des sémaphores (semaphore.h, semaphore.c).

Indication :

Les deux processus se synchronisent en utilisant des sémaphores. Au début le processus P2 est bloqué, P1 affiche sa première série débloque le P2 et se bloque. P2 affiche sa première série débloque le P1 et se bloque, et ainsi de suite. Pour la mémoire partagée, reprendre l'exercice 3 de la série 4.