

TD Programmation Système

SMI S6

Série 4 : Mémoires Partagées

Exercice 1 : Création Mémoires Partagées

Deux processus P1 et P2 sans liens parenté partagent un segment de mémoire permettant de stocker des réels. Le processus P1 crée la Mémoires Partagée et l'initialise avec la valeur passée en paramètre. Le deuxième processus P2 accède au segment récupère la valeur calcule son carrée et affiche le résultat. Ecrire les deux programmes C sous Unix implémentant P1 et P2.

Appels Systèmes : shmget(), shmat(), shmctl(),shmdt()

Exercice 2 :

Deux processus P1 et P2 sans liens parenté partagent un segment de mémoire de type entier. On désire avoir l'affichage suivant a l'exécution

P1 : 2 4 6 8 10

P2 : 13 16 19 22 25

Ecrire les deux programmes C sous Unix implémentant P1 et P2.

Appels Systèmes : shmget(), shmat(), shmctl(),shmdt()

Indication :

Le processus P1 crée la mémoire partagée l'utilise pour afficher les multiples de 2 jusqu'au premier multiple de 5 et il arrête son exécution cède la main au processus P2 qui accède à la mémoire récupère la dernière valeur l'utilise pour afficher les multiples de 3 jusqu'au premier multiple de 5.

Exercice 3 : Synchronisation

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.

Indication :

Les deux processus se synchronisent par l'envoi de message en utilisant la fonction kill(). Chacun doit avoir connaissance du pid de l'autre, une mémoire partagée et alors nécessaire pour pouvoir stocker les deux pid.