

SMI S6

Programmation Système

**Exercice I : Cours**

1. Dans l'appel système `fork()` comment un processus accède au pid du père ? comment le père accède au pid du fils ?
2. Quel est la différence entre les appels systèmes `shmget()` et `malloc()` ?
3. Quel est la différence entre les appels systèmes `wait()` et `waitpid()` ?
4. Quel moyen utilise l'appel `shmget()` pour garantir le partage d'un segment de données entre plusieurs processus ?
5. Quelle est la différence entre un pipe nommé et un fichier ?
6. Pour quelle raison la communication dans un pipe est unidirectionnelle ?

**Exercice II : Mémoire Partagée et Sémaphores**

Soit la structure suivante :

```
typedef struct MP{  
    int tab[10];  
    int compteur;} MP;
```

1. On désire partager la structure MP par deux processus P1 et P2. Donner la portion de code permettant de créer et d'attacher la mémoire partagée MP par les deux processus.
2. La structure MP permet-elle de partager les pid des deux processus ? Si non donner sa nouvelle syntaxe.
3. On désire contrôler l'exécution concurrente des deux processus en respectant les synchronisations suivantes :
  - Au départ le processus P1 est bloqué
  - Le processus P2 initialise les 5 premières cases du tableau (tab) et il se bloque
  - Le processus P1 reprend son exécution, initialise les 5 dernières cases du tableau (tab) et termine son exécution
  - Le processus P2 reprend son exécution en affichant la somme des éléments du tableau (tab)

**Question :**

Ecrire le code C de P1 et P2 permettant de gérer la synchronisation en utilisant les sémaphores.  
On suppose qu'on dispose des fonctions de gestion des sémaphores définies dans `semaphore.h`  
`Creat_Sem(int cle), Init_Sem(int id, int valeur), void P(int id), void V(int id)`