

---

Méthodes et langages pour le parallélisme

---

---

Travaux pratiques

---

Exercice 1

Ecrire les fonctions relatives aux sémaphores dans un fichier semaphore.c, écrire aussi semaphore.h.

```
int P(int numEns, int numSem, int nbjetons)
int V(int numEns, int numSem, int nbjetons)
int creerSem(key-t KEY, int Droits)
int ouvrirSem(key-t KEY)
int initialiserSem(int nens, int numSem, int Value)
int detruireSem(int numEns)
```

Compiler uniquement ces fichiers avec la commande gcc -c

Exercice 2

Nous allons au cours de cet exercice tester les fonctions écrites à l'exercice 1

Pour cela concevoir une application où deux processus affichent à tour de rôle à l'écran

Vous pouvez faire un père et un fils

Ou deux processus indépendants cad 2 mains

Exercice 3

Variante 0 du producteur consommateur classique

On se propose d'étudier une variante du modèle des producteurs/consommateurs dans lequel on supposera que plusieurs processus producteurs et plusieurs processus consommateurs coexistent.

On suppose que tous les messages sont mémorisés séquentiellement dans un tampon de N cases gérées circulairement.

Exercice 4

Dans cette variante, deux types de messages existent toujours. Les producteurs sont différenciés par le type des messages qu'ils déposent dans le tampon. Contrairement à la variante 2, les dépôts dans le tampon se font dans l'ordre des arrivées, quel que soit le type du message déposé. En revanche, les consommateurs précisent quel type de message ils désirent retirer. Les messages sont toujours retirés dans l'ordre des dépôts et les consommateurs sont traités dans l'ordre de leurs arrivées.