TP 4 – Les Sémaphores

L'objectif de ce TP est d'introduire les sémaphores de la norme POSIX. Pour cela vous allez utiliser une bibliothèque standard qui permet de les manipuler, à savoir la bibliothèque «**semaphore.h**». Cette bibliothèque inclut les fonctions suivantes :

```
#include <semaphore.h>
int sem_init(sem_t *sem, int pshared , unsigned int valeur);
int sem_wait(sem_t * sem);
int sem_post(sem_t * sem);
int sem_destroy(sem_t *sem);
```

- **sem_init** initialise le sémaphore pointé par *sem*. Le compteur associé au sémaphore est initialisé à *valeur*.
  L'argument *pshared* indique si le semaphore est partagé entre processus. Ceci n'est pas valable dans les threads de Linux. L'argument *pshared* sera toujours positionné à zéro.
- **sem_wait** suspend le processus appelant, le compteur du sémaphore est automatiquement décrémenté (**Primitive P**). Renvoie toujours **0**.
- **sem_post** incrémente automatiquement le compteur du sémaphore pointé par *sem*. Cette fonction ne bloque jamais (**Primitive V**). Renvoie **0** en cas de succès et **-1** en cas d'erreur.
- int **sem_ destroy**(sem_t *sem) détruit l'espace occupé par le sémaphore. Renvoie **0** en cas de succès et **-1** en cas d'erreur.

Exercice 1 :

Écrire un programme qui permet de créer deux threads partageant une variable «note». Le premier Thread incrémente de 1 la note. Le deuxième Thread décrémente de 1 la note.

Compléter ce programme pour résoudre le problème de l'EM à l'aide des sémaphores.

(Remarque : Il est toujours préférable d'utiliser les mutex pour résoudre l'EM)

Exercice 2 :

Un enseignant souhaite réaliser un programme lui permettant de gérer les notes de son module.

Il crée 3 Threads réalisant les tâches suivantes :

Le 1er Thread calcule la note de TD: *TD=examTD+présence.*

Le 2ème Thread calcule la note de TP: *TP=examTP+présence.*

La note de présence est la même pour le TD et le TP et est représentée par une seule variable.

Le 3ème Thread calcule la note du module *note=(TD+TP)/2.*

Écrire le programme réalisant ces tâches tout en assurant leur synchronisation.

Exercice 3 : Adapter le programme précédent aux tâches suivantes (ajouter les sémaphores de synchronisation et d'EM nécessaires)

P1 calcule *TD=TD+examTD+Présence* P2 calcule *TD=TD+2 ;*

P3 calcule TP=*examTP+Présence* P4 calcule *note=(TD+TP)/2 ;*