

Implémentation des sémaphores avec les conditions

On souhaite implémenter les sémaphores en utilisant les « conditions » offertes par l'API POSIX des threads.

On rappelle que les sémaphores sont constitués d'un compteur et de deux opérations P et V, dont l'implémentation peut être :

P	V
<hr/>	<hr/>
compteur--	compteur++
si compteur < 0	si compteur ≤ 0
alors attendre	alors réveiller un thread
fin si	fin si

Les algorithmes ci-dessus doivent être implémentés en section critique.

- définissez le type `monsem_t` nécessaire pour représenter un sémaphore;
- implémentez la fonction `int monsem_init (monsem_t *, int)`, où le deuxième paramètre est la valeur initiale du compteur;
- implémentez la fonction `int monsem_P (monsem_t *)`;
- implémentez la fonction `int monsem_V (monsem_t *)`.

Comme toutes les fonctions `sem_*`, ces fonctions doivent renvoyer 0 si tout s'est bien passé ou -1 en cas d'erreur (et modifier la variable `errno`).

Le programme `ex3th.c` donne une base pour tester vos fonctions.

À l'aide des sémaphores que vous avez implémentés, écrivez un programme `cat2th` composé de deux threads : le thread généré lit des octets sur l'entrée standard et les place un par un dans un tableau de 10 éléments que le thread principal lit et affiche sur la sortie standard.

Votre programme doit bien évidemment :

- ne comporter aucune variable globale;
- s'arrêter lorsqu'une fin de fichier est rencontrée sur l'entrée standard;
- fonctionner sans problème avec des données binaires;
- produire une sortie identique à l'entrée; par exemple, la commande suivante doit afficher « ok ».

```
$ ./cat2th < /bin/ls > toto && cmp /bin/ls toto && echo ok
```