

## T.P 1

Ce Tp de manipulation de sémaphore s'effectuera avec un système de sémaphores simple nomme EZSEM. Vous trouverez donc les fichiers ezsem.h, ezsem.c, logfile.h et logfile.c dans ezsem.tar.gz. Un fichier Makefile est de plus donné pour vous rappeler son utilisation.

Voici son fonctionnement :

Le programme principal doit contenir une ligne : `const char *LOGF_TITLE_APPLI = "nom_de_l'application";`  
pour initialiser le système de sémaphores, utiliser :  
`EZSEM_INIT(FILE *logfile, 0)`

On dispose de 10 sémaphores repérés par un entier entre 0 et 9  
appel `EZSEM_INIT_COUNTER(i,k)` pour initialiser le compteur du sémaphore `i` avec la valeur entière  $k \geq 0$  à faire AVANT tout emploi du sémaphore `i`  
appel `EZSEM_P(i)` ou `EZSEM_V(i)`  
appel `EZSEM_END` pour fermer le système de sémaphores (attend qu'aucun autre processus que la racine ne manipule les sémaphores : non indispensable mais si non appelé, les processus gérant les sémaphores meurent après la fin du processus racine.)  
Après la fin d'exécution de `EZSEM_END`, il faut un délai (quelques secondes) pour que les processus de gestion des sémaphores soient détruits.

Quelques informations supplémentaires sur la fonction `wait(&status)` : pour récupérer proprement un code de retour d'exit, certaines macro sont indispensables pour extraire les bits de poids forts et faibles du code. Voici quelques macros :

- `WEXITSTATUS(status)` : renvoie le code de retour du processus fils passé à exit,
- `WIFEXITED(status)` : renvoie vrai si le statut provient d'un processus fils qui s'est terminé en quittant le main avec `return` ou avec un appel à exit.

Un *man wait* vous donnera les autres.

### Exercice 1

Écrire un programme qui crée 3 processus écrivant je suis le processus `i` ( $i=1, 2$  ou  $3$ ).

Ajouter des sémaphores pour être certain de toujours obtenir : je suis le processus 1, je suis le processus 2, je suis le processus 3. Combien de sémaphores faut-il ? Donner le schéma illustrant les processus ainsi que les sémaphores.

### Exercice 2 (le diner de philosophes chinois)

Le problème est le suivant : 5 philosophes sont autour d'une table pour se restaurer. Chacun possède une baguette et une seule!!! Le philosophe étant bien élevé ne mange qu'avec deux baguettes, il doit donc en emprunter une à un voisin.

**Question 0.1** Comment modéliser ce problème avec des sémaphores ? Donner une solution et l'implanter.

**Question 0.2** Vous avez trouvé une solution qui fonctionne et vous l'avez "prouvé" ? Très bien jeune padawan, trouvez une solution à ce problème via des tubes.