

# Exercices sur les variables de condition

## Exercice 1

Comment peut-on implémenter une barrière de synchronisation réutilisable en utilisant seulement des variables de condition ?

Veillez respecter la sémantique des barrières POSIX: une fois tous les threads passés, la barrière est réinitialisée à sa valeur initiale.

Voici la structure de barrière à compléter :

```
typedef struct {  
    // complete please  
} barrier_t;
```

Et ci-dessous les fonctions à implémenter :

```
void barrier_init(barrier_t *bar, int count) {  
    // complete please  
}  
  
void barrier_wait(barrier_t *bar) {  
    // complete please  
}  
  
void barrier_destroy(barrier_t *bar) {  
    // complete please  
}
```

Bien entendu, il vous faudra aussi développer un programme de test permettant de vérifier que votre implémentation fonctionne correctement. Encore une fois, vous devriez pouvoir remplacer n'importe quel code utilisant les barrières POSIX avec votre implémentation à base de variables de condition.

Par exemple, utilisez cette barrière à la place de la barrière POSIX de la série d'exercice précédente. Vous devriez trouver les mêmes résultats.

## Exercice 2

Nous désirons écrire un programme modélisant l'accès à un pont en utilisant les variables de condition. Des voitures et des camions désirent passer sur ce pont. Celui-ci est un peu fragile et ne peut supporter qu'un certain poids. Les contraintes du système sont les suivantes :

- Le système à modéliser ne se termine jamais : les voitures et les camions font un tour en campagne pendant un certain temps, puis tentent de traverser le pont.
- Il y a deux types de véhicules : des camions et des voitures.
- Le nombre total de voitures et de camions est arbitraire et passé en paramètre du programme.
- Les temps de balade d'un véhicule en campagne, ainsi que le temps mis pour traverser le pont, sont tous deux aléatoires.
- La charge maximale supportée par le pont doit être spécifiée en argument du programme.
- Les masses des camions et des voitures doivent toutes deux être passées en argument de votre programme.
- Chaque véhicule est modélisé par un thread.

Le but de cet exercice est d'implémenter, un algorithme d'accès au pont en vérifiant naturellement que le poids total des véhicules roulant sur le pont ne dépasse jamais la charge maximale de celui-ci. Si cette condition n'est pas satisfaite, tout véhicule doit s'arrêter (i.e. être bloqué) à l'entrée du pont. La masse d'une voiture, d'un camion, ainsi que la charge maximale du pont sont des paramètres à spécifier sur la ligne de commande du programme.