Théorie et pratique de la concurrence – Master 1 Informatique
TP 3 : Verrous et variables de condition en C

Les variables de condition

La bibliothèque `Pthreads` propose également un autre outil de synchronisation entre processus que l'on appelles les variables de condition. Une variable de condition est déclarée comme suit :

```
pthread_cont_t (varcond);
```

Pour l'initialiser, on procède ainsi :

```
pthread_cond_init (&vacond,NULL);
```

Pour utiliser les variables conditionnelles, on dispose de trois fonctions qui doivent être appelées au sein d'une section critique créée grâce à un *mutex*. La première fonction bloque le *thread* appelant et libère le verrou. Lorsque le *thread* bloqué est ensuite réveillé, il reprend la main sur le verrou. Elle s'appelle de la façon suivante :

```
pthread_cond_wait (&varcond,&verrou);
```

Le deuxième argument correspond au `mutex` utilisé pour la section critique. Un code utilisant cette fonction aura donc l'allure suivante :

```
pthread_mutex_lock(&verrou);
pthread_cond_wait (&varcond,&verrou);
pthread_mutex_unlock(&verrou);
```

Pour réveiller un *thread* bloqué sur une variable de condition, un autre thread peut faire :

```
pthread_cond_signal(&varcond);
```

Là aussi il est **recommandé** d'appeler cette fonction au sein d'une section critique protégée par le *mutex* utilisé par le thread en attente (dans l'exemple précédent il s'agit du *mutex* `verrou`). On peut aussi réveiller tous les *threads* en attente grâce à la fonction :

```
pthread_cond_broadcast(&verrou);
```

Exercices

Exercice 1: *Les philosophes*

Faire l'exercice 4 du TP précédent.

Exercice 2: *Mémoire partagée simple*

Reprendre l'exercice 4 du Tp1 en utilisant les variables de conditions pour attendre si le flag est mis à vrai ou faux.

Exercice 3: *Problème du bus*

Nous considérons N passagers et un bus ayant C places (avec $C < N$). Le comportement du bus est le suivant :

(a) Il attend que C passagers soient montés.

(b) Il part

(c) Il attend que les C passagers soient descendus du bus pour aller de nouveau au point (a).

Le comportement d'un passager est le suivant :

(a') Il essaie de monter dans le bus, si il y a encore de la place, il peut monter sinon il doit attendre.

(b') Une fois monté dans le bus, il fait le voyage dans le bus.

(c') Il descend du bus et retourne en (a').

Représenter ce système avec des processus concurrents utilisant des verrous et des variables de condition.

Indication : Voilà une façon possible d'aborder le problème :

- Les passagers pourront à l'aide d'un compteur partagé se compter de façon à savoir combien de passagers sont montés dans le bus et combien sont descendus. (Attention à protéger l'accès à ce compteur partagé grâce à un sémaphore binaire).
- Le dernier passager à remplir le bus réveillera le bus pour lui signaler qu'il est plein.
- Le dernier passager à descendre du bus réveillera le bus pour lui signaler qu'il est vide.

Remarques : Observer différentes exécutions et dire si un passager voulant monter dans le bus n'y arrive jamais. Si c'est le cas, savez-vous d'où vient ce problème ? Si ce n'est pas le cas, pouvez expliquer pourquoi ?