

Programmation Concurrente

Exclusion mutuelle par attente active

Exercice 1

Est-ce que l'algorithme ci-dessous garanti l'exclusion mutuelle (exclusion mutuelle, progression, attente bornée) entre les threads T0 et T1 ? Si ce n'est pas le cas, justifiez votre réponse en décrivant une séquence possible d'opérations qui prouve le contraire.

```
bool inside[2] = {false, false};

// id is the thread number (0 or 1)
void prelude(int id) {
    while (inside[1-id]) {}
    inside[id] = true;
}

void postlude(int id) {
    inside[id] = false;
}
```

Exercice 2

Le compilateur gcc offre la fonction suivante qui est garantie d'être atomique :

```
int __sync_val_compare_and_swap(int *ptr, int oldval, int newval);
```

Cette fonction implémente, de façon **atomique**, l'équivalent des instructions ci-dessous :

```
int compare_and_swap(int *reg, int oldval, int newval) {
    int old_reg_val = *reg;
    if (old_reg_val == oldval)
        *reg = newval;
    return old_reg_val;
}
```

Nous aimerions réaliser un mécanisme d'exclusion par attente active à l'aide de la fonction :

`__sync_val_compare_and_swap`.

Voici le squelette de ce que nous aimerions implémenter :

```
// Cette fonction initialise le verrou passé en argument
void init_lock(<type_a_definir> *lock_ptr) {
    // A remplir
}
```

```
// Cette fonction verrouille (acquiert) le verrou passé en
argument
void acquire_lock(<type_a_definir> *lock_ptr) {
    // A remplir
}
```

```
// Cette fonction déverrouille (relâche) le verrou passé en
argument
void release_lock(<type_a_definir> *lock_ptr) {
    // A remplir
}
```

Pour cet exercice, il est demandé :

1. D'implémenter le code ci-dessus (structure et fonctions)
2. De vérifier que l'implémentation fonctionne correctement en verrouillant un compteur incrémenté par 2 threads de manière concurrente.

Exercice 3

Implémentez l'algorithme de Peterson pour l'exclusion mutuelle par attente active dans le cadre de 2 threads.

Elaborez ensuite un scénario afin de tester que votre implémentation fonctionne correctement.

- Que remarquez-vous ?
- Si le résultat que vous obtenez n'est pas celui escompté, essayez de trouver un moyen pour corriger le problème.