

LINFO1252 – Systèmes informatiques

Travail Dirigé (TD) Semaine 9 : Analyser un algorithme d'exclusion mutuelle

Objectifs de ce TD : S'entraîner à prouver des propriétés sur un algorithme.

Compétences travaillées : preuve d'algorithmes, raisonnement par l'absurde

Prérequis : Avoir lu le syllabus sur la mise en œuvre de la synchronisation,

<https://sites.uclouvain.be/SystInfo/notes/Theorie/Threads/spinlocks.html>

Durée du TD : 40 minutes. Le reste de la séance est consacré au projet P1.

Considérons l'algorithme d'exclusion mutuelle publié par Harris Hyman en 1966 dans le « journal de l'ACM », un périodique scientifique. Cette solution vise deux threads T0 et T1.

```
1  int flag[2] = {0, 0};
2  int turn = 0;
3
4  void lock(int mon_id) {
5      flag[mon_id] = 1;
6      while (turn != mon_id) {
7          while (flag[1 - mon_id] == 1) {}
8          turn = mon_id;
9      }
10 }
11
12 void unlock(int mon_id) {
13     flag[mon_id] = 0;
14 }
```

Les deux threads T0 et T1 appellent de façon infinie la séquence :

```
1  while (true) {
2      lock(I); // I = 0 ou 1 selon le thread
3      section_critique();
4      unlock(I); // I = 0 ou 1 selon le thread
5  }
```

En utilisant les méthodes employées pendant le cours pour analyser différents algorithmes d'exclusion mutuelle, vous devez montrer que (si) cet algorithme :

- Respecte l'exclusion mutuelle (propriété de safety)
- Ne mène pas à une situation de livelock ou un autre problème de liveness

Pour rappel, pour prouver qu'une propriété est fausse un contre-exemple suffit (par exemple une suite d'instructions exécutées par T0 et T1). Pour prouver qu'une propriété est vraie dans tous les cas il est souvent utile d'utiliser un raisonnement par l'absurde.