
Exercice de programmation concurrente: Gestion d'une section critique

On considère l'algorithme (en pseudo-code) ci-dessous dont l'objectif est d'implémenter une section critique pour deux threads numérotés 0 et 1. La variable i correspond au numéro du thread.

```
1
2  /* initialisation */
3  bool flag[2] = {FALSE,FALSE}; /* tableau de booléens */
4  int turn = 0;
5
6  flag[i]= TRUE;
7  while (turn != i) {
8      while (flag[1-i]) {
9          /* corps de boucle vide */
10     }
11     turn = i;
12 }
13
14 ... /* section critique */
15
16 flag[i] = FALSE; /* sortie de section critique */
```

Cette solution est incorrecte. Il est demandé :

- de préciser quelle est la propriété qui n'est plus respectée (exclusion mutuelle, progrès, attente bornée);
- d'expliquer/illustrer le problème avec un exemple de séquence d'exécution problématique.