

## TP 3 (2023) Systèmes concurrents et distribués

**Exercice 1 :** L'algorithme de Kessel ci-dessous est une solution du problème d'exclusion mutuelle pour deux processus.

- Montrez qu'il résout le problème de l'exclusion mutuelle, c'est-à-dire que jamais deux threads ne se trouvent en section critique simultanément.
- Montrez que l'algorithme est starvation et deadlock free.

Utilisez le modèle d'entrelacement mais limitez vous à considérer les entrelacements des variables partagées. Remarquez que les variables *b* et *turn* sont partagées, les variables *local* sont locales

Initialement :  $b = \{\text{false}, \text{false}\}$ ,  $\text{turn} = \text{quelconque}$

### Thread 0

```
b[0]=true;
local[0]=turn[1];
turn[0]=local[0];
while ((b[1]==true && (local[0]==turn[1])))
```

critical section

```
b[0] = false;
```

### Thread 1

```
b[1]=true;
local[1]=1-turn[0];
turn[1]=local[1];
while ((b[0]==true && (local[1]!=turn[0])))
```

critical section

```
b[1] = false;
```

### Exercice 2 : Suite de l'exercice 3 TP2.

Testez l'algorithme de Peterson pour vous convaincre qu'il est bien solution du problème de la section critique. Testez l'algorithme en

déclarant les variables partagées avec le mot clé **volatile**. Que se passe-t-il si vous l'omettez ?