

TP 2 (2019) Systèmes concurrents et distribués

Exercice 1 :

Implémentez les tentatives 1,2 et 3 (pages 51-70) de l'interface Lock (p.54) et utilisez un programme de test (pages 57-58) pour générer des exécutions.

Essayez l'implémentation de l'algorithme de Peterson sans déclarer les variables partagées de type **volatile**. Observez si le protocole est toujours correct ?

Exercice 2 : Implémentez le tampon vu au cours page 101 (voir ci-dessous). Exécutez le programme avec un thread consommateur (qui appelle deq) et un thread producteur (qui appelle enq).

Testez le programme avec deux threads consommateurs et un thread producteurs.

Remplacez dans vos code enq() et deq() l'instruction notifyAll() par notify() et QSIZE=1. Avec 2 consommateurs et 1 producteur observez un interblocage. Proposez une exécution qui résulte en un interblocage.

```

public class queue {
    private int head = 0, tail = 0;
    final int QSIZE=1;
    Item[QSIZE] items;

    public synchronized void enq(Item x) {
        while (this.tail – this.head == QSIZE)
            this.wait();
        this.items[this.tail++%QSIZE] = x;
        this.notify();
    }

    public synchronized Item deq(Item x) {
        while (this.tail – this.head == 0)
            this.wait();
        Item inter = this.items[this.head%QSIZE];
        this.head+=1;
        this.notify();
        return inter;
    }
}

```