

**Haute école d'ingénierie et de gestion du Canton de Vaud**  
**Département TIC**  
**Laboratoire de programmation répartie PRR**

Temps à disposition : 10 périodes

Distribué le : mardi 30 octobre 2018 à 14h55

A rendre le : mardi 4 décembre 2018 à 14h55 (avant la séance de labo)

---

## Objectifs

- Comprendre le fonctionnement d'un algorithme d'exclusion mutuelle réparti.
- Etendre l'algorithme de Lamport pour maintenir la cohérence d'une variable répartie sur un ensemble de sites.
- Réaliser des communications par Java RMI.

## Énoncé

Partager une donnée parmi un ensemble de sites est un problème qui peut se résoudre par le biais d'un algorithme d'exclusion mutuelle. Dans ce laboratoire, nous souhaitons modifier l'utilisation prévue de l'algorithme de Lamport afin de résoudre ce problème.

Chaque site détient une variable entière qui doit être cohérente. Les tâches applicatives peuvent faire 2 opérations sur cette variable : **consulter sa valeur**, et **modifier sa valeur**. La consultation revient à obtenir la valeur la plus récente. Par contre, une modification se passe en 3 étapes :

1. obtenir l'exclusion mutuelle sur la variable;
2. modifier la valeur en section critique, par exemple incrémenter sa valeur;
3. informer tous les autres sites de la nouvelle valeur quand la section critique est libérée.

## Travail à faire

Chaque site de l'environnement exécute 2 machines virtuelles. La première est un gestionnaire qui maintient la cohérence de la variable entière sur le site. C'est cette VM qui implémente l'algorithme de Lamport et qui détient la variable globale. La seconde VM exécute les tâches applicatives. Ces tâches applicatives fournissent un menu permettant d'effectuer les 2 opérations suivantes :

1. afficher la valeur de la variable globale obtenue par une requête au gestionnaire du site (opération de consultation);
2. obtenir la permission de modifier la variable globale, puis réalisation de cette modification par une saisie au clavier.

La première opération n'est jamais bloquante et pourra être effectuée par toutes les tâches applicatives qui ne sont ni en attente d'une exclusion mutuelle et ni en cours de modifier la variable globale. C'est uniquement la seconde opération qui nécessite l'obtention de l'exclusion mutuelle, et afin de vérifier le fonctionnement de votre implémentation, vous devez afficher la valeur de la variable globale en section critique avant sa modification. Aussi, après la libération de la section critique, la nouvelle valeur de la variable globale sera jointe au message de libération de Lamport de manière à informer les autres sites de sa nouvelle valeur.

## Hypothèses

1. Le nombre de sites partageant la donnée n'est pas connu, **mais peut être introduit comme paramètre sur la ligne de commande**. Une fois initialisé, le nombre de sites demeure constant.
2. Ces sites et le réseau qui les interconnecte sont entièrement fiables (ne tombent jamais en panne).
3. La communication entre les différentes VM se réalisera uniquement par RMI.
4. Afin de simplifier l'initialisation des gestionnaires d'exclusion mutuelle, vous êtes libre de prendre toutes les hypothèses qui simplifient le problème. Typiquement, vous devez lancer

toutes les VM maintenant la cohérence de la variable globale avant de lancer les tâches applicatives.

### Remarques sur le travail à effectuer

- Vous devez nous rendre un listage (papier) complet de vos sources (fichiers sources Java).
- La description de l'implémentation, ses différentes étapes et toute autre information pertinente doivent figurer dans les programmes rendus. Aucun rapport n'est demandé.
- Inspirez-vous du barème de correction pour connaître là où il faut mettre votre effort.
- Vous pouvez travailler en équipe de deux personnes.
- Une démo de votre programme pourrait être demandée.

### Barème de correction

Conception (structure, décomposition, conformité)	20%
Portabilité sur un autre environnement réparti, réutilisation du code dans un projet plus conséquent, robustesse	10%
Exécution et fonctionnement (démo)	20%
Codage (choix des variables, opérations, lisibilité, localité de référence, etc.)	15%
Documentation des en-têtes (programme et méthodes)	25%
Commentaires au niveau du code (qualité et complétude)	10%