

TP noté de conception de protocoles réseaux informatiques

Modélisation de l'algorithme d'exclusion mutuelle de Lamport

Le but du TP est de réaliser un modèle formel de l'algorithme distribué d'exclusion mutuelle de Lamport en utilisant des horloges de Lamport implémentées en langage Promela et l'interpréteur Spin¹ vu en cours. Cet algorithme permet de garantir qu'un et un seul processus entre en section critique dans un système distribué de processus communiquant par message.

Les requêtes d'entrée en section critique sont exécutées dans l'ordre croissant des estampilles de Lamport : les requêtes avec une estampille ayant une valeur la plus petite recevront la permission d'entrer en section critique avant celle ayant une estampille plus grande.

Le protocole contient 3 types de messages : (**REQUEST**, **REPLY** et **RELEASE**), les canaux de communications sont de type FIFO, sans perte, ré-ordonnancement ou duplication.

- Un processus envoie un message **REQUEST** à tous les autres processus pour obtenir leur permission d'entrer dans la section critique.
- Un processus réponds avec un message **REPLY** pour autoriser le processus qui le demande d'entrer dans la section critique.
- Un processus envoie un message **RELEASE** à tous les autres processus lorsqu'il quitte la section critique.
- Chaque processus P_i maintient une file d'attente des requêtes d'entrée en section critique, ordonnée par leur estampille de Lamport. **request_queue_i** est la file d'attente du process P_i
- Une estampille est donnée à chaque message de requête en utilisant une horloge de Lamport.
- Les estampilles permettent d'ordonner la priorité d'entrée en section critique de manière distribuée. Les estampilles les plus petites sont priorisées sur les estampilles les plus grande. L'exécution des sections critiques est le même que l'ordre indiqué par les estampilles.

Algorithme:

- **Pour entrer en section critique :**
 - Lorsqu'un processus P_i veut entrer en section critique, il envoie un message de requête **Request(tp_i, i)** à tous les autres processus et met la requête dans une file d'attente de requête **request_queue_i**. Tp_i est l'estampille de Lamport du processus P_i
 - Lorsqu'un processus P_j reçoit le message de requête **REQUEST(tp_j, i)** du process P_i , il lui renvoie un message **REPLY** estampillé au processus P_i **ET** copie la requête du processus P_i dans **request_queue_j** .
- **Pour exécuter la section critique:**
 - Un processus P_i peut entrer dans la section critique si il a reçu un message avec une estampille plus grande que **(tp_j, i)** de tous les autres processus **ET** que sa requête est la première dans **request_queue_i**
- **Pour libérer la section critique:**

¹ <http://spinroot.com/spin/whatispin.html>

- Lorsqu'un processus P_i veut libérer la section critique, il retire sa requête de **request_queue_i** et envoie un message **RELEASE** estampillé à tous les autres processus.
- Lorsqu'un processus P_j reçoit un message **RELEASE** message du processus S_i , il enlève la requête de P_i de **request_queue_j**.

Partie 1 : Horloge de lamport

Vous implémenterez les horloges de Lamport telles que vue en cours dans le langage Promela².

A la fin de l'exécution de votre modèle, un ordre total des évènements classés par les horloges et lamport devra être affiché sur la console.

Important : Votre modèle des horloges de Lamport devra fonctionner avec un nombre quelconque de processus paramétrable dans le code source de votre modèle.

Partie 2 : Algorithme d'exclusion mutuelle basé sur les horloges de Lamport

Sur la base de la partie 1. Vous implémenterez l'algorithme d'exclusion mutuelle de Lamport.

Le format des messages et le nombre de canaux de votre implémentation est libre.

A l'aide de l'instruction `assert` et du vérificateur exhaustif de Spin³, vous démontrerez que l'exclusion mutuelle est assurée sur un nombre prédéfini de processus (au moins 3).

Modalité de rendu et d'évaluation

- Le rendu sous la forme du code source promela déposé sur un dépôt git et dont l'URL sera communiquée au professeur. Le TP noté est à réaliser en binôme.
- Le code source, **abondamment commenté**, devra contenir la modélisation de l'algorithme d'exclusion mutuelle de Lamport telle que décrit dans ce document.
- La partie 1 devra être rendu avant le **4/12/2020 23h59**. La partie 2 devra être rendue avant le **18/12/2020 23h59**. La partie deux sera évaluée par un oral d'explication et de démonstration du modèle.

² https://hepia.infolibre.ch/reseauxI-2020-2021/lamport_clocks.pdf

³ https://hepia.infolibre.ch/reseauxI-2020-2021/verifying_a_promela_model.pdf