

LOG8470

Méthodes formelles en fiabilité et en sécurité

TP 2 : Élection dynamique d'un leader

Session	Automne 2021
Pondération	5 % de la note finale
Taille des équipes	2 étudiants
Date de remise	5 octobre (23 h 55 au plus tard)
Directives particulières	Soumission des fichiers Promela par moodle dans un dossier zip sous la forme tp2-matricule1_matricule2.zip . Toute soumission du livrable en retard est pénalisé à raison de 10 % par jour de retard.
Moyen de communication	<ul style="list-style-type: none">- Serveur discord- Courriel : basil.capar@polymtl.ca

1. Notion du cours à connaître

- Élection dynamique d'un *leader*

2. Objectif

Ce laboratoire est séparé en 2 exercices. L'objectif du premier est d'implanter l'algorithme de Dolev-Klawe-Rodeh. Le second exercice a pour but d'implanter le *Bully algorithm*.

3. Utilité de l'élection dynamique d'un *leader*

L'élection dynamique d'un *leader* permet de sélectionner un nœud qui coordonne un système au complet. Effectivement, ce processus est souvent utilisé dans les systèmes distribués, ce qui permet d'assigner des tâches à différents nœuds, modifier des données et de s'occuper des différentes requêtes qui sont passées dans le système. En effet, lorsque les bases de données sont répliquées, on peut déterminer un *leader* qui s'occupera d'effectuer des écritures.

Le partage de ressources dans un système distribué est assez complexe¹. Effectivement, on peut penser au fait que les systèmes distribués sont semblables à des *threads* qui possèdent un processus, mais tous ces *threads* sont des ordinateurs qui parviennent à se communiquer grâce au réseau. Ainsi, il n'y a pas de principe de sémaphores et de mutex pour l'utilisation des ressources partagées². À cet effet, Dolev-Klawe-Rodeh et *Bully* sont deux des algorithmes qui permettent de remédier à ces problèmes.

¹ [Implementing leader election on Google Cloud Storage | Google Cloud Blog](#)

² [Leader election in distributed systems \(amazon.com\)](#)

4. Algorithme de Dolev-Klawe-Rodeh (3 pts)

Ce premier exercice consiste à expliquer comment l'algorithme Dolev-Klawe-Rodeh fonctionne et de l'implémenter. Au niveau de l'algorithme, il est important de savoir que les nœuds communiquent grâce à des *channels*. Seul 6 nœuds sont nécessaires pour représenter le système distribué. Vous devez considérer qu'il n'y a pas eu de crash parmi les nœuds et que c'est la première fois que le système distribué se met en marche. Ainsi, parmi les 6 nœuds, on veut déterminer lequel sera le leader pour que le système soit opérationnel. Bien entendu, le résultat final que vous devrez obtenir est 6 qui est le id du leader.

5. Bully algorithm (1,5 pts)

Quant à l'algorithme *Bully*, il faut également l'expliquer et l'implémenter. Contrairement à l'exercice ci-dessus, quatre nœuds sont suffisants pour représenter l'algorithme. Aussi, il faut savoir que vous devez implémenter cet algorithme en sachant que le *leader* était le nœud #4, mais il n'est plus opérationnel et que c'est le nœud #1 qui a détecté la panne. Ainsi, il faut que le système trouve un nouveau *leader* qui est le nœud #3.

Les nœuds doivent se communiquer grâce aux *channels*. Aussi, les messages de communications possibles sont *Error*, *Election*, *Ok* et *Coordinator*. Bien entendu, les nœuds 1, 2 et 3 sont opérationnels. Cependant, chacun de ces nœuds devra s'assurer que les autres nœuds sont fonctionnels en vérifiant qu'ils n'ont pas reçu de message *Error*.

6. Discussion entre les 2 algorithmes (0,5 pt)

Discuter des différences entre les deux algorithmes.

7. Bonus (0,5 pt)

Implémentez un algorithme personnalisé qui permet d'élire un leader dynamiquement et expliquez comment il fonctionne.

8. Modalité de remise

Vous devez remettre un fichier *zip* qui contient les 2 fichiers *pml* ainsi qu'un rapport. Le rapport devra être nommé comme suit : tp2-matricule1_matricule2.pdf. Pour les fichiers promela, ils devront être nommés comme suit : tp2-bully-matricule1_matricule2.pml et tp2-dolev-matricule1_matricule2.pml.

9. Critères d'évaluation

Qualité du code	1.5 points
Utilisation des canaux pour communiquer entre les processus	1 point
Le <i>leader</i> élu est le bon	1 point
Implémentation adéquate des algorithmes <i>Bully</i> et Dolev-Klawe-Rodeh	1.5 points
Utilisation des goto	-1 point
Code non exécutable	-3 points
Mauvais format de remise	-1 point
Retard	-10 % par jour de retard