

L'algorithme d'exclusion mutuelle de Lamport

L'avantage de cet algorithme d'exclusion mutuelle est qu'en cas d'absence de demande de section critique, le système finit par atteindre un état dans lequel plus aucun message ne circule.

Hypothèses :

- Processus et canaux asynchrones.
- Canaux FIFO.
- Processus avec identité.
- Pas de faute.
- Topologie en clique. (Au moins deux nœuds.)
- Multi-initiateurs

Principe : L'algorithme est défini par les cinq règles suivantes. Chacune de ces règles sera considérée comme un unique événement.

1. Pour demander la section critique, un processus p envoie le message $\langle \text{demande}, t_m \rangle$ à tous les autres processus où t_m est la valeur de l'horloge de p au moment de l'envoi et met ce message dans sa liste Requête.
2. Lorsqu'un processus p reçoit un message $\langle \text{demande}, t_m \rangle$ du processus q , il met ce message dans sa liste Requête et envoie un accusé réception Ack à q tagué avec sa valeur d'horloge.
3. A la fin de sa section critique, le processus p envoie un message $\langle \text{sortie}, t_m \rangle$ à tous les autres processus où t_m est la valeur de l'horloge de p au moment de l'envoi.
4. Lorsqu'un processus p reçoit un message $\langle \text{sortie}, t_{mi} \rangle$ du processus q , il supprime toute requête de q présente dans sa liste Requête.
5. Un processus p accède à sa section critique quand il y a une requête (t_m, p) dans sa liste Requête telle que :
 - a. (t_m, p) est plus petite que toutes les autres requêtes de Requête suivant l'ordre.
 - b. p a reçu un message tagué avec une valeur d'horloge supérieure à t_m de chacun des autres processus.

Ecrivez un algorithme appliquant les règles précédentes puis démontrez qu'il répond aux spécifications de notre problème.