WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools
What links here
Related changes
Upload file
Special pages
Permanent link
Page information
Wikidata item
Cite this page

Print/export
Create a book
Download as PDF
Printable version

Languages
Add links

Lamport's distributed mutual exclusion algorithm

From Wikipedia, the free encyclopedia
(Redirected from Lamport's Distributed Mutual Exclusion Algorithm)

It has been suggested that this article be merged into *Lamport's bakery algorithm*. (Discuss) *Proposed since October 2013.*

Lamport's Distributed Mutual Exclusion Algorithm is a contention-based algorithm for mutual exclusion on a distributed system.

Contents [hide]

1 Algorithm
1.1 Nodal properties
1.2 Algorithm
2 Message complexity
3 Drawbacks
4 See also
5 References

Algorithm [edit]

Nodal properties [edit]

1. Every process maintains a queue of pending requests for entering critical section in order. The queues are ordered by virtual time stamps derived from Lamport timestamps.[1]

Algorithm [edit]

Requesting process

1. Enters its request in its own queue (ordered by time stamps)
2. Sends a request to every node.
3. Wait for replies from all other nodes.
4. If own request is at the head of its queue and all replies have been received, enter critical section.
5. Upon exiting the critical section, remove its request from the queue and send a release message to every process.

Other processes

1. After receiving a request, enter the request in its own request queue (ordered by time stamps) and reply with a time stamp.
2. After receiving release message, remove the corresponding request from its own request queue.
3. If own request is at the head of its queue and all replies have been received, enter critical section.

Message complexity [edit]

This algorithm creates $3(N − 1)$ messages per request, or $(N − 1)$ messages and 2 broadcasts. $3(N − 1)$ messages per request includes:

- $(N − 1)$ total number of requests
- $(N − 1)$ total number of replies
- $(N − 1)$ total number of releases

Drawbacks [edit]

There exist multiple points of failure.

See also [edit]

- Ricart-Agrawala algorithm (an improvement over Lamport's algorithm)

- Lamport's Bakery Algorithm
- Raymond's Algorithm
- Maekawa's Algorithm
- Suzuki-Kasami's Algorithm
- Naimi-Trehel's Algorithm

References [edit]

1. ^ Kshemkalyani, A., & Singhal, M. Chapter 9: Distributed Mutual Exclusion Algorithms. Distributed Computing: Principles, Algorithms, and Systems (Page 10 of 93). Cambridge University Press.

This computer science article is a stub. You can help Wikipedia by expanding it.

Categories: Computer science stubs | Concurrency control algorithms