

7 Lamport's algorithm for distributed database.

In a distributed database system, multiple sites (nodes) may need to access a shared resource such as a data item, file or critical section (CS).

Since there is no shared clock or shared memory coordination is difficult.

→ Objectives

1. Mutual exclusion
2. No deadlock
3. No starvation
4. Fairness.

→ The algorithm is fair in the sense that a request for CS are executed in the order of their timestamp and time is determined by logical clocks.

→ When a site processes a request for the CS, it updates its local clock and assigns the request a timestamp.

→ The algorithm executes CS requests in the increasing order of timestamps.

→ Every site s_i keeps a queue, request queue which contains mutual exclusion request ordered by their timestamp.

→ Algorithm execute CS request in the increasing order of timestamp.

Algorithm

1. Requesting the critical section

- When a site s_i wants to enter the CS, it broadcasts a REQUEST (ts_i, i) message to all other sites and places the request on request-queue, (ts_i, i) denotes the timestamp of the request.

- When a site s_j receives the REQUEST (ts_i, i) message from site s_i , it places site s_i 's request in request-queue and

returns a timestamped REPLY message to s_i .

2 Executing the critical section

Site s_i enters the CS when the following two conditions hold:

L1: s_i has received a message with timestamp larger than (ts_i, i) from all other sites.

L2: s_i 's request is at the top of request_queue.

Releasing the critical section

- Site s_i , upon exiting the CS, removes its request from the top of its request queue and broadcasts a timestamp RELEASE message to all other sites.

- When a site s_j receives a RELEASE message from site s_i it removes s_i 's request from its request queue.

Realtime Application of Lamport's Algorithm

1. Distributed database system

- * Controlling concurrent access to shared database records
- * Ensuring transaction consistency.

2. Distributed File System

- * Synchronizing access to shared files
- * Preventing write conflicts.

3. cloud computing Environment

- * Managing access to shared cloud resources.
- * Coordinating distributed services.

4. Distributed operating systems

- * Scheduling critical section execution across nodes

5. Replication management.

Requirements to implement the algorithm

1. Reliable Message passing - without loss and Finite.
2. Logical clock mechanism - Each process must maintain and update Lamport timestamp
3. Request Queue - A priority queue sorted by (timestamp, Process id)
4. Stable Network
5. Knowledge of other process
6. Unique process identifiers.