

# PRESENTATION

By group 8

## **8.BULLY ALGORITHM FOR DISTRIBUTED DATABASE.**

The Bully Algorithm is used in a distributed system to elect a coordinator (leader) when the current coordinator fails.

# **NEED AND IMPORTANCE OF THE BULLY ALGORITHM IN A DISTRIBUTED DATABASE SYSTEM**

A distributed database system consists of multiple interconnected nodes that store and process data collaboratively. To maintain consistency, reliability, and coordination, one node usually acts as a coordinator (leader).

# **NEED AND IMPORTANCE OF THE BULLY ALGORITHM IN A DISTRIBUTED DATABASE SYSTEM**

The coordinator is responsible for tasks such as:

- Transaction coordination
- Concurrency control
- Deadlock handling
- Commit protocols (e.g., Two-Phase Commit)
- Synchronization among nodes

# **NEED AND IMPORTANCE OF THE BULLY ALGORITHM IN A DISTRIBUTED DATABASE SYSTEM**

Since distributed systems are prone to node failures, the coordinator may crash or become

unreachable. If no replacement is chosen quickly, the system may face:

- Transaction failures
- Data inconsistency
- Deadlocks
- Reduced availability

# **NEED AND IMPORTANCE OF THE BULLY ALGORITHM IN A DISTRIBUTED DATABASE SYSTEM**

The Bully Algorithm is important because it provides a systematic and automatic way to elect a new coordinator when the current one fails. It ensures that the most capable process (highest priority/ID) becomes the leader, thereby maintaining efficient control over the distributed database.

## DESCRIPTION OF THE BULLY ALGORITHM

The Bully Algorithm is a leader election algorithm used in distributed systems where each process has a unique priority number (ID). The process with the highest ID is always elected as the coordinator.

# DESCRIPTION OF THE BULLY ALGORITHM

## ASSUMPTIONS

- Every process has a unique ID.
- All processes know the IDs of other processes.
- Message delivery is reliable.
- A failed coordinator can be detected using timeouts.
- Higher ID processes have higher priority.

## WORKING OF THE ALGORITHM

- When a process detects that the coordinator is not responding, it initiates an election.
- The process sends an ELECTION message to all processes with higher IDs.
- If no higher-ID process responds, the initiating process declares itself the coordinator

## WORKING OF THE ALGORITHM

- If a higher-ID process responds with an OK message, the initiating process stops the election.
- The higher-ID process then starts its own election.
- This continues until the highest active process is found.
- The elected process broadcasts a COORDINATOR message to all other processes.

# **REAL-TIME APPLICATIONS OF THE BULLY ALGORITHM**

The Bully Algorithm is used in systems where quick and reliable leader election is required.

# REAL-TIME APPLICATIONS:

## **1. Distributed Database Management Systems**

- Selecting a coordinator for transaction management and commit protocols.

## **2. Distributed File Systems**

- Choosing a master node for metadata management.

## **3. Cluster Computing**

- Leader election in server clusters to manage workload distribution.

# REAL-TIME APPLICATIONS:

## 4. Fault-Tolerant Systems

- Replacing failed controllers in real-time systems.

## 5. Cloud Computing Environments

- Electing a master node among virtual machines for orchestration tasks.

## 6. Microservices Architecture

- Selecting a leader for configuration management or service discovery.

## REQUIREMENTS TO IMPLEMENT THE BULLY ALGORITHM IN A DISTRIBUTED SYSTEM

- ***To successfully implement the Bully Algorithm, the following requirements must be met:***

# **REQUIREMENTS TO IMPLEMENT THE BULLY ALGORITHM IN A DISTRIBUTED SYSTEM**

## ***System Requirements:***

- Multiple interconnected nodes (distributed environment)
- Reliable communication network
- Failure detection mechanism (timeouts or heartbeat messages)

# **REQUIREMENTS TO IMPLEMENT THE BULLY ALGORITHM IN A DISTRIBUTED SYSTEM**

## ***Software Requirements:***

- Message-passing mechanism
- Process ID assignment logic
- Election, OK, and Coordinator message handlers
- Timeout and retry mechanisms

# **REQUIREMENTS TO IMPLEMENT THE BULLY ALGORITHM IN A DISTRIBUTED SYSTEM**

## ***Algorithmic Requirements:***

- Algorithmic Requirements:
- Unique and comparable process IDs
- Knowledge of all active processes
- Ability to detect coordinator failure
- Capability to broadcast messages

# **REQUIREMENTS TO IMPLEMENT THE BULLY ALGORITHM IN A DISTRIBUTED SYSTEM**

## ***Hardware Requirements:***

- Networked computers or servers
- Stable communication links

# THANK YOU

Any Questions?