Bully Algorithm

Roll number 1–5

01

Need and Importance

The Bully Algorithm is used in distributed systems to elect a coordinator (leader) among multiple processes or nodes. In a distributed system, many computers work together, and a coordinator is required to manage tasks such as resource allocation, synchronization, and communication control.

- Helps in leader election when the current coordinator fails or crashes.
- Ensures the system continues to function properly without manual intervention.
- Allows higher priority processes to take control, improving efficiency.
- Maintains fault tolerance by automatically selecting a new coordinator.
- Supports system reliability and stability in distributed environments.

02

Description of the Algorithm

1. When a process detects that the coordinator is not responding, it starts an election.
2. The process sends an election message to all processes having higher IDs.
3. If no higher-ID process responds, the current process becomes the coordinator and sends a coordinator message to all other processes.
4. If a higher-ID process responds, that process takes over the election process.
5. The process with the highest ID eventually becomes the coordinator.

03

Real-time Application of the Algorithm

- Distributed database systems for selecting master database servers.
- Cloud computing platforms for selecting a primary server.
- Network management systems for selecting control nodes.
- Distributed IoT systems for selecting central monitoring devices.
- Multi-server web applications for load balancing and coordination.

04

Requirements to Implement the Algorithm in the System

1. Each process must have a unique identification number.
2. All processes should know about other processes in the system.
3. Reliable communication between processes is required.
4. The system must support failure detection mechanisms.
5. Processes must be able to send and receive election and coordinator messages.
6. The system should allow dynamic joining and leaving of processes.

Thank you