Leader Election Using Bully Algorithm - Project Report

Group Members: **Faiza Kashaf and Anas Naeem**

Course: **Distributed Computing**

Instructor: **Dr Muhammad Saeed**

This project demonstrates the Bully Algorithm, a leader election protocol used in distributed systems to select a coordinator among multiple processes. Each process has a unique ID, and the one with the highest ID is selected as the coordinator. If the coordinator fails, a new election is triggered by any active node. The algorithm ensures a new leader is elected even if some nodes are down.

The implementation is done in **C** and provides a terminal-based interface with improved UI using color-coded messages and clean menus. It allows the user to activate/deactivate nodes, initiate elections, and display current node status including the active coordinator.

Key Features:

- Colored terminal UI using ANSI codes

- Dynamic node management (activate/fail)

- Recursive election handling based on process IDs

- Realistic simulation of failure and recovery in distributed systems

The project is a simplified simulation and aims to help students understand fault tolerance and coordination in networked environments.

Compile: **gcc bully.c -o bully**

To run the Program: **./bully**

PROBLEMS   OUTPUT   DEBUG CONSOLE   **TERMINAL**   PORTS   bash + ∨ ⊓ ⬚ ··· ∧ ✕

Enter number of nodes: ^[[A^[[A^C
● faiza@faiza-Lenovo-V14-G3-IAP:~/Downloads/Bully Algorithm Implementation$ gcc bully.c -o bully
● faiza@faiza-Lenovo-V14-G3-IAP:~/Downloads/Bully Algorithm Implementation$ ./bully
----
    LEADER ELECTION USING BULLY ALGORITHM (C Simulation)

Enter number of nodes: 5
Enter unique IDs for 5 nodes:
2 4 6 8 9

Initial Coordinator is Node with ID 9
----
1. Fail a Node
2. Activate a Node
3. Start Election from a Node
4. Display Node Status
5. Exit

Enter your choice: 1
Enter node index to fail (0 to 4): 4
Node 9 is now inactive.
Coordinator has failed. Start an election.
----
1. Fail a Node
2. Activate a Node
3. Start Election from a Node
4. Display Node Status
5. Exit

EXPLORER   C bully.c   ✕
∨ BULLY ALGORITHM IMPLEMENTATION   C bully.c
  ≡ bully   56   int main() {
  C bully.c   95   case 1:
                                        96   printf("Enter node index to fail (0 to %d): " n - 1);

PROBLEMS   OUTPUT   DEBUG CONSOLE   **TERMINAL**   PORTS   bash + ∨ ⊓ ⬚ ···

Enter index of node to start election (0 to 4): 1

Election initiated by Node 1 (ID 4)...
→ Node 2 (ID 6) responds to election.
→ Node 3 (ID 8) responds to election.

Election initiated by Node 3 (ID 8)...

✓ Node 3 (ID 8) becomes the new Coordinator.
----
1. Fail a Node
2. Activate a Node
3. Start Election from a Node
4. Display Node Status
5. Exit

Enter your choice: 4

Node Status:
Node 0 (ID: 2) : Active
Node 1 (ID: 4) : Active
Node 2 (ID: 6) : Active
Node 3 (ID: 8) : Active
Node 4 (ID: 9) : Inactive

Current Coordinator: Node with ID 8

> OUTLINE
> TIMELINE