Department of Computer Engineering
Academic Term : January - May 2024

Class: B.E. (Computer) Semester VIII Division: B

Subject Name: Distributed Systems CSC 801

Name	**Riya Rajesh Patil**
Roll No.	**9282**
Experiment No.	**5**
Experiment Title	**To implement Election Algorithms.**

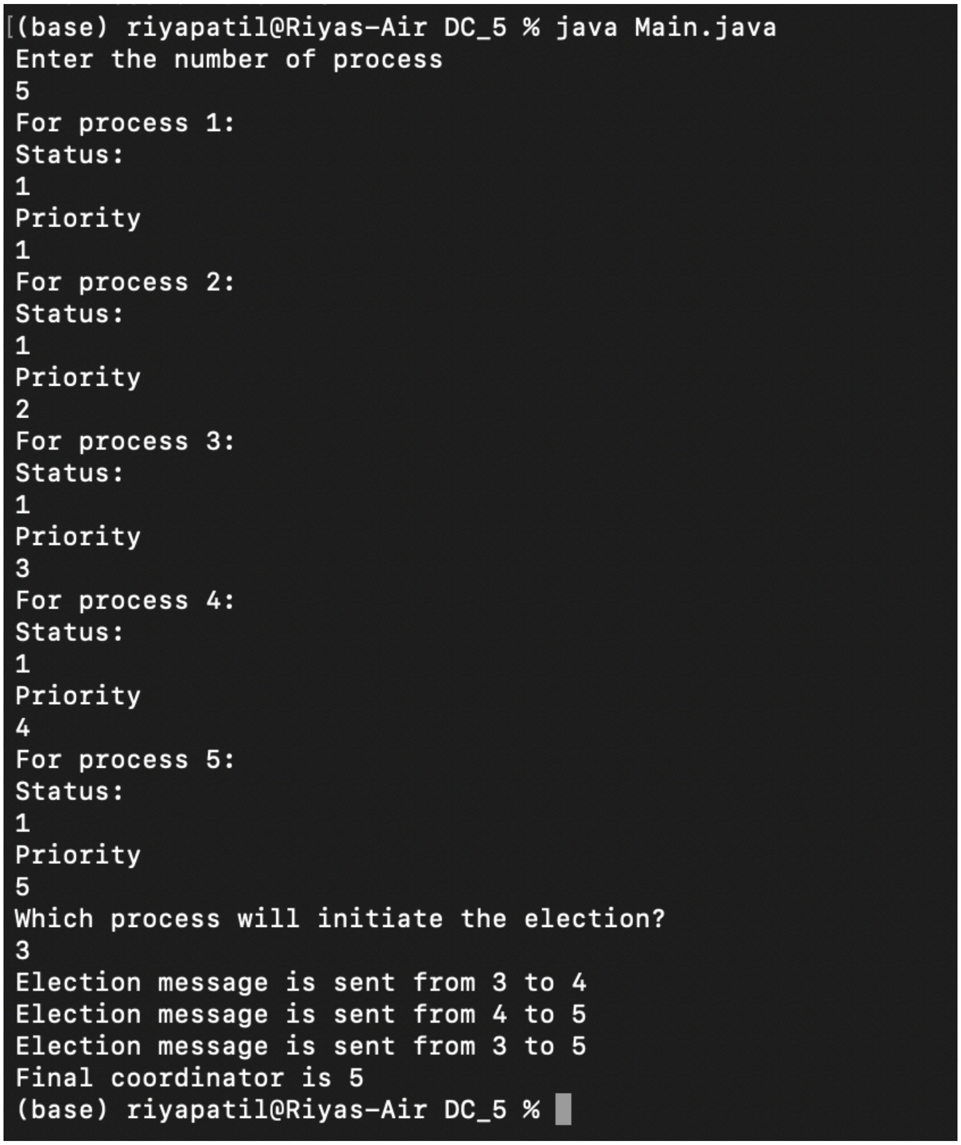
Aim - To implement Election Algorithms.

Main.java

```
import java.io.*;
import java.util.Scanner;
class Main {
static int n;
static int pro[] = new int[100];
static int sta[] = new int[100];
static int co;
public static void main(String args[]) throws IOException {
System.out.println("Enter the number of process");
Scanner in = new Scanner(System.in);
n = in.nextInt();
int i;
for (i = 0; i < n; i++) {
System.out.println("For process " + (i + 1) + ":");
System.out.println("Status:");
sta[i] = in.nextInt();
System.out.println("Priority");
pro[i] = in.nextInt();
}
System.out.println("Which process will initiate the election?");
int ele = in.nextInt();
elect(ele);
System.out.println("Final coordinator is " + co);
}
static void elect(int ele) {
```

```
ele = ele - 1;
co = ele + 1;
for (int i = 0; i < n; i++) {

if (pro[ele] < pro[i]) {
System.out.println("Election message is sent from " + (ele + 1) + " to " + (i + 1));
if (sta[i] == 1)
elect(i + 1);
}
}
}
}
```



```
[(base) riyapatil@Riyas-Air DC_5 % java Main.java
Enter the number of process
5
For process 1:
Status:
1
Priority
1
For process 2:
Status:
1
Priority
2
For process 3:
Status:
1
Priority
3
For process 4:
Status:
1
Priority
4
For process 5:
Status:
1
Priority
5
Which process will initiate the election?
3
Election message is sent from 3 to 4
Election message is sent from 4 to 5
Election message is sent from 3 to 5
Final coordinator is 5
(base) riyapatil@Riyas-Air DC_5 %
```

Conclusion -

Here I can conclude that the bully election algorithm is useful to elect the new leader when the current leader fails. From the output it is seen that the node with the highest priority will eventually become the coordinator node. In my output I chose five Processes. Process 1 has status 1 meaning that it is active along with the priority 1.Process 2 has status 1 and priority 2. Process 3 has status 1 and priority 3.Process 4 has status 1 and priority 4.Process 5 has status 1 and priority 5. For any process if the status is 0 it means that it is not active. Process 3 will initiate the election. The bully election algorithm will run over all the processes and finally process 5 will get elected as the coordinator as it has the highest priority.

Postlab -

1. **Give some examples where you need a coordinator in the Distributed System.**
   In distributed systems, coordinators play a crucial role in coordinating and managing various activities to ensure the system operates smoothly. Here are some examples where coordinators are needed:

   1. Leader Election : In systems with a leader-follower architecture, such as distributed message brokers or distributed file systems, a coordinator is necessary to facilitate leader election. The coordinator helps nodes determine which node should act as the leader for a given period.

   2. Distributed Locking: Coordinators are used to manage distributed locks in scenarios where multiple nodes need to access shared resources. The coordinator coordinates lock acquisition and release to prevent conflicts and ensure data consistency.

   3. Distributed Task Scheduling: In distributed computing frameworks like Apache Spark or Hadoop, a coordinator is responsible for scheduling tasks across multiple nodes. It decides which tasks should be executed on which node based on factors like resource availability and data locality.

   4. Load Balancing: In distributed systems handling incoming requests, a coordinator can act as a load balancer, distributing requests among multiple nodes to ensure optimal resource utilization and prevent overload on any single node.

2. **What is the significance of middle ware?**
   Middleware acts as a vital connector in modern computing, simplifying the complexities of distributed systems. It enables easier development, deployment, and management of applications across various environments by offering crucial services like messaging, data management, and security. By abstracting away infrastructure details and providing standardized interfaces, middleware fosters scalability, flexibility, and resilience.

Additionally, it enhances system efficiency through features like load balancing and fault tolerance, simplifying development challenges in distributed computing.