**Filename: BullyAlgorithm.java**

import java.util.\*;

public class BullyAlgorithm {

private int numProcesses;

private boolean[] processes;

private int leader;

public BullyAlgorithm(int numProcesses) {

this.numProcesses = numProcesses;

this.processes = new boolean[numProcesses];

Arrays.fill(processes, true);

this.leader = numProcesses - 1;

}

public void startElection(int failedProcess) {

System.out.println("Process " + failedProcess + " has failed!");

for (int i = failedProcess + 1; i < numProcesses; i++) {

if (processes[i]) {

System.out.println("Sending election message from " + i + " to " + leader);

if (sendMessage(i, leader)) {

leader = i;

}

}

}

System.out.println("Process " + leader + " is the new leader!");

}

public boolean sendMessage(int source, int destination) {

System.out.println("Sending message from " + source + " to " + destination);

return processes[destination];

}

public static void main(String[] args) {

int numProcesses = 5;

BullyAlgorithm algorithm = new BullyAlgorithm(numProcesses);

algorithm.startElection(2);

}

}

---------------------------------------------------------------------------------------------------------------------------

**Filename: RingAlgorithm.java**

import java.util.\*;

public class RingAlgorithm {

private int numProcesses;

private int[] processes;

private int leader;

public RingAlgorithm(int numProcesses) {

this.numProcesses = numProcesses;

this.processes = new int[numProcesses];

for (int i = 0; i < numProcesses; i++) {

processes[i] = i;

}

this.leader = numProcesses - 1;

}

public void startElection(int failedProcess) {

System.out.println("Process " + failedProcess + " has failed!");

int sender = (failedProcess + 1) % numProcesses;

boolean leaderFound = false;

while (!leaderFound) {

System.out.println("Sending election message from " + sender + " to " + (sender + 1) % numProcesses);

int receiver = (sender + 1) % numProcesses;

if (sendMessage(sender, receiver)) {

sender = receiver;

} else {

for (int i = 0; i < numProcesses; i++) {

if (i != sender && sendMessage(sender, i)) {

sender = i;

}

}

leaderFound = true;

}

}

System.out.println("Process " + leader + " is the new leader!");

}

public boolean sendMessage(int source, int destination) {

System.out.println("Sending message from " + source + " to " + destination);

if (destination == leader) {

return true;

}

return false;

}

public static void main(String[] args) {

int numProcesses = 5;

RingAlgorithm algorithm = new RingAlgorithm(numProcesses);

algorithm.startElection(2);

}

}