

Assignment - 6

1. Bully Algorithm

Code :

```
import java.util.Scanner;

class Process{

    public int id;
    public String status;

    public Process(int id){
        this.id = id;
        this.status = "active";
    }
}

public class BullyAlgo{

    Scanner sc;
    Process[] processes;
    int n;

    public void ring(){

        sc=new Scanner(System.in);

        System.out.println("Enter Number of Processes");
        n=sc.nextInt();

        processes = new Process[n];

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

            processes[i] = new Process(i);
        }
    }

    public void performElection(){
```

```

int max_id = getMax();

System.out.println("Process having highest Process Id "+ max_id + " falis ");

processes[max_id].status = "inactive";

int initiator = 0;

while(true){

    boolean higher_process = false;

    for(int i=initiator+1; i<n; i++){

        if(processes[i].status=="active"){

            System.out.println("Process "+initiator+ " Passes Election message to
process "+i);
            higher_process=true;

        }

    }

    System.out.println();

    if(higher_process){

        for(int i=initiator+1; i<n; i++){

            if(processes[i].status=="active"){

                System.out.println("Process "+i+" Passes Ok ("+"i+") message to
process" + initiator);
            }

        }

        initiator++;
        System.out.println();
    }
    else{

        int coord = getMax();
        System.out.println("Finally Process "+coord+" Becomes Coordinator");

        System.out.println();
    }
}

```

```

        for(int i = coord - 1; i >= 0; i--){

            if(processes[i].status == "active"){

                System.out.println("Process "+coord+" Passes Coordinator (" + coord + ")
message to process " + i);
            }
        }

        System.out.println();

        System.out.println("End of Election");

        break;
    }
}

```

```

public int getMax(){

    int id=-1;
    int max_pid=-999;

    for(int i=0; i<processes.length; i++){

        if(processes[i].status=="active" && max_pid < processes[i].id){

            max_pid = processes[i].id;
            id=i;
        }
    }

    return id;
}

```

```

public static void main(String[] args) {

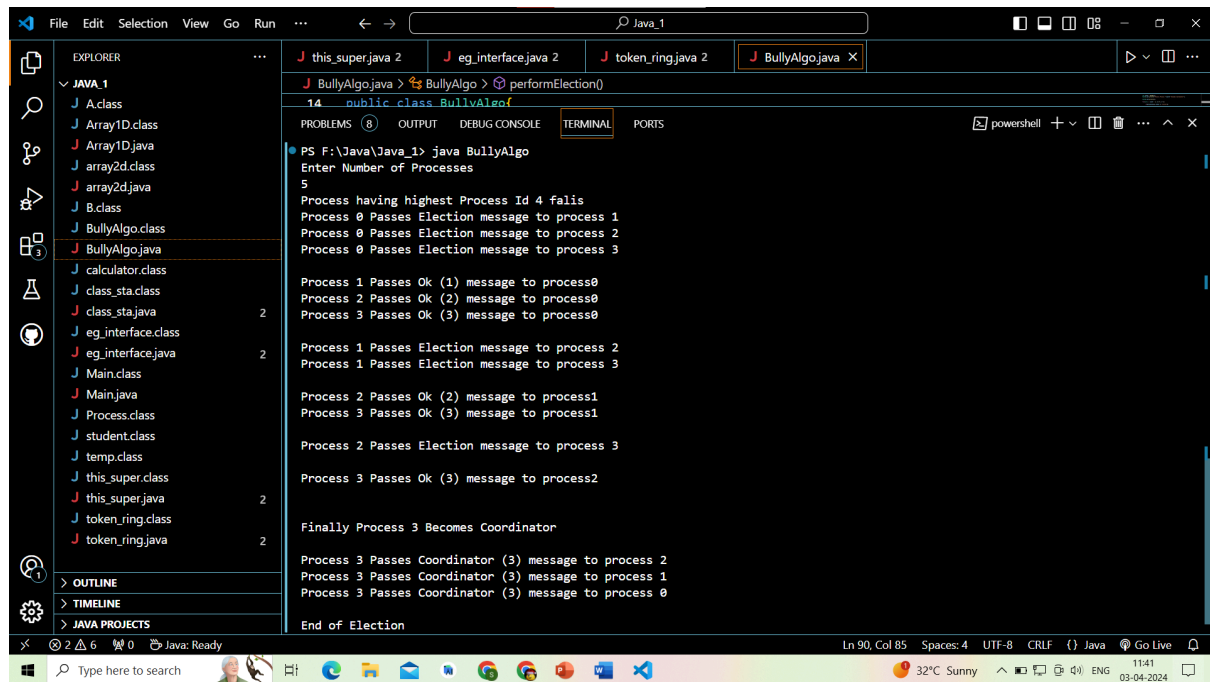
    BullyAlgo bully = new BullyAlgo();

    bully.ring();
    bully.performElection();

}}

```

Output :



```
public class BullyAlgo {
    public void performElection() {
        // ... (omitted code) ...
        System.out.println("Finally Process 3 Becomes Coordinator");
        // ... (omitted code) ...
    }
}
```

PS F:\Java\Java_1> java BullyAlgo
Enter Number of Processes
5
Process having highest Process Id 4 fails
Process 0 Passes Election message to process 1
Process 0 Passes Election message to process 2
Process 0 Passes Election message to process 3

Process 1 Passes Ok (1) message to process0
Process 2 Passes Ok (2) message to process0
Process 3 Passes Ok (3) message to process0

Process 1 Passes Election message to process 2
Process 1 Passes Election message to process 3

Process 2 Passes Ok (2) message to process1
Process 3 Passes Ok (3) message to process1

Process 2 Passes Election message to process 3

Process 3 Passes Ok (3) message to process2

Finally Process 3 Becomes Coordinator

Process 3 Passes Coordinator (3) message to process 2
Process 3 Passes Coordinator (3) message to process 1
Process 3 Passes Coordinator (3) message to process 0

End of Election

2. Bully Ring Algorithm

Code :

```
import java.util.*;

public class Ring_Bully {
    int max_processes;
    int coordinator;
    boolean processes[];
    ArrayList<Integer> pid;

    public Ring_Bully(int max) {

        coordinator = max;
        max_processes = max;
        pid = new ArrayList<Integer>();
        processes = new boolean[max];

        for(int i = 0; i < max; i++) {
            processes[i] = true;
            System.out.println("P" + (i+1) + " created.");
        }
        System.out.println("P" + (coordinator) + " is the coordinator");
    }
}
```

```

}

void displayProcesses() {
    for(int i = 0; i < max_processes; i++) {
        if(processes[i])
            System.out.println("P" + (i+1) + " is up.");
        else
            System.out.println("P" + (i+1) + " is down.");
    }
    System.out.println("P" + (coordinator) + " is the coordinator");
}

void upProcess(int process_id) {

    if(!processes[process_id-1]) {

        processes[process_id-1] = true;
        System.out.println("Process P" + (process_id) + " is up.");
    } else {
        System.out.println("Process P" + (process_id) + " is already up.");
    }
}

void downProcess(int process_id) {
    if(!processes[process_id-1]) {
        System.out.println("Process P" + (process_id) + " is already down.");
    } else {
        processes[process_id-1] = false;
        System.out.println("Process P" + (process_id) + " is down.");
    }
}

void displayArrayList(ArrayList<Integer> pid) {
    System.out.print("[ ");
    for(Integer x : pid) {
        System.out.print(x + " ");
    }
    System.out.print("]\n");
}

void initElection(int process_id) {

    if(processes[process_id-1]) {
        pid.add(process_id);

        int temp = process_id;

        System.out.print("Process P" + process_id + " sending the following list:- ");

```

```

displayArrayList(pid);

while(temp < max_processes && temp != process_id - 1) {
    if(processes[temp]) {
        pid.add(temp+1);
        System.out.print("Process P" + (temp + 1) + " sending the following list:- ");
        displayArrayList(pid);
    }
    temp = (temp + 1) % max_processes;
}
coordinator = Collections.max(pid);
System.out.println("Process P" + process_id + " has declared P" + coordinator + " as
the coordinator");
pid.clear();
}
}

```

```

public static void main(String args[]) {

```

```

    Ring_Bully ring = null;
    int max_processes = 0, process_id = 0;
    int choice = 0;
    Scanner sc = new Scanner(System.in);

```

```

while(true) {
    System.out.println();
    System.out.println("Ring Algorithm");
    System.out.println("1. Create processes");
    System.out.println("2. Display processes");
    System.out.println("3. Up a process");
    System.out.println("4. Down a process");
    System.out.println("5. Run election algorithm");
    System.out.println("6. Exit Program");
    System.out.print("Enter your choice:- ");
    choice = sc.nextInt();

```

```

switch(choice) {
    case 1:
        System.out.print("Enter the total number of processes:- ");
        System.out.println();
        max_processes = sc.nextInt();
        ring = new Ring_Bully(max_processes);
        break;
    case 2:
        System.out.println();
        ring.displayProcesses();
        break;
    case 3:

```


File Edit Selection View Go Run ... Java_1

EXPLORER

- Java_1
 - A.class
 - Array1D.class
 - Array1D.java
 - array2d.class
 - array2d.java
 - B.class
 - BullyAlgo.class
 - BullyAlgo.java
 - calculator.class
 - class_sta.class
 - class_sta.java
 - eg_interface.class
 - eg_interface.java
 - Main.class
 - Main.java
 - Process.class
 - Ring_Bully.class
 - Ring_Bully.java
 - student.class
 - temp.class
 - this_super.class
 - this_super.java
 - token_ring.class
- OUTLINE
- TIMELINE
- JAVA PROJECTS

J this_super.java 2 J eg_interface.java 2 J token_ring.java 2 J BullyAlgo.java J Ring_Bully.java 1

J Ring_Bully.java > Ring_Bully > initElection(int)

```
4 public class Ring_Bully {
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Ring Algorithm

1. Create processes
2. Display processes
3. Up a process
4. Down a process
5. Run election algorithm
6. Exit Program

Enter your choice:- 5
Enter the process which will initiate election:- 5

Process P5 sending the following list:- [5]
Process P5 has declared P5 as the coordinator

Ring Algorithm

1. Create processes
2. Display processes
3. Up a process
4. Down a process
5. Run election algorithm
6. Exit Program

Enter your choice:- 6
PS F:\Java\Java_1> []

Ln 63, Col 9 Spaces: 4 UTF-8 CRLF {} Java Go Live

Type here to search 36°C Sunny 14:29 03-04-2024