# Assignment No: 6

**Title / Objective:** Bully and Ring Algorithm

**Statement:** Implement Bully and Ring algorithm for leader election.

**Course Outcome:** C414454.2

**Requirements:** Java Programming Environment, JDK 1.8, Eclipse Neon(EE).

**Input:**

**Bully.java**

```
package ass6;
import java.util.Scanner;

public class Bully {
	static boolean[] state = new boolean[5];
	int coordinator;

	public static void up(int up) {
		if (state[up - 1]) {
			System.out.println("process" + up + "is already up");
		} else {
			int i;
			Bully.state[up - 1] = true;
			System.out.println("process " + up + "held election");
			for (i = up; i < 5; ++i) {
				System.out.println("election message sent from process" + up + "to process" +
(i + 1));
			}
			for (i = up + 1; i <= 5; ++i) {
				if (!state[i - 1])
					continue;
				System.out.println("alive message send from process" + i + "to process" + up);
				break;
			}
		}
	}

	public static void down(int down) {
		if (!state[down - 1]) {
			System.out.println("process " + down + "is already dowm.");
		} else {
			Bully.state[down - 1] = false;
		}
	}

	public static void mess(int mess) {
		if (state[mess - 1]) {
			if (state[4]) {
				System.out.println("0K");
			} else if (!state[4]) {
				int i;
				System.out.println("process" + mess + "election");
```

```
                        for (i = mess; i < 5; ++i) {
                                System.out.println("election send from process" + mess + "to process "
+ (i + 1));
                        }
                        for (i = 5; i >= mess; --i) {
                                if (!state[i - 1])
                                        continue;
                                System.out.println("Coordinator message send from process" + i + "to
all");
                                break;
                        }
                }
        } else {
                System.out.println("Prccess" + mess + "is down");
        }
    }

    public static void main(String[] args) {
        int choice;
        Scanner sc = new Scanner(System.in);
        for (int i = 0; i < 5; ++i) {
                Bully.state[i] = true;
        }
        System.out.println("5 active process are:");
        System.out.println("Process up = p1 p2 p3 p4 p5");
        System.out.println("Process 5 is coordinator");
        do {
                System.out.println(".....");
                System.out.println("1 up a process.");
                System.out.println("2.down a process");
                System.out.println("3 send a message");
                System.out.println("4.Exit");
                choice = sc.nextInt();
                switch (choice) {
                        case 1: {
                                System.out.println("bring proces up");
                                int up = sc.nextInt();
                                if (up == 5) {
                                        System.out.println("process 5 is co-ordinator");
                                        Bully.state[4] = true;
                                        break;
                                }
                                Bully.up(up);
                                break;
                        }
                        case 2: {
                                System.out.println("bring down any process.");
                                int down = sc.nextInt();
                                Bully.down(down);
                                break;
                        }
                        case 3: {
                                System.out.println("which process will send message");
                                int mess = sc.nextInt();
                                Bully.mess(mess);
```

```
                        }
                    }
                } while (choice != 4);
        }
}
```

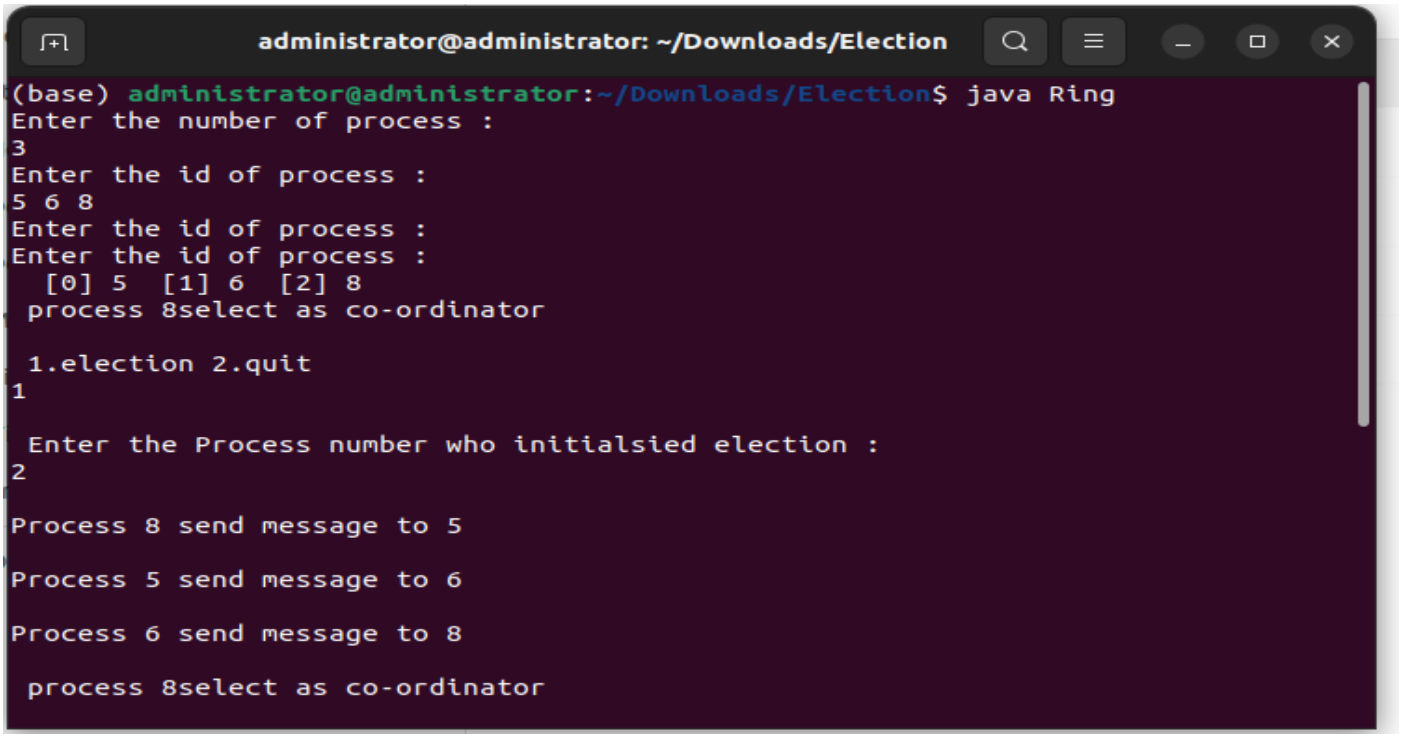
**Ring.java**

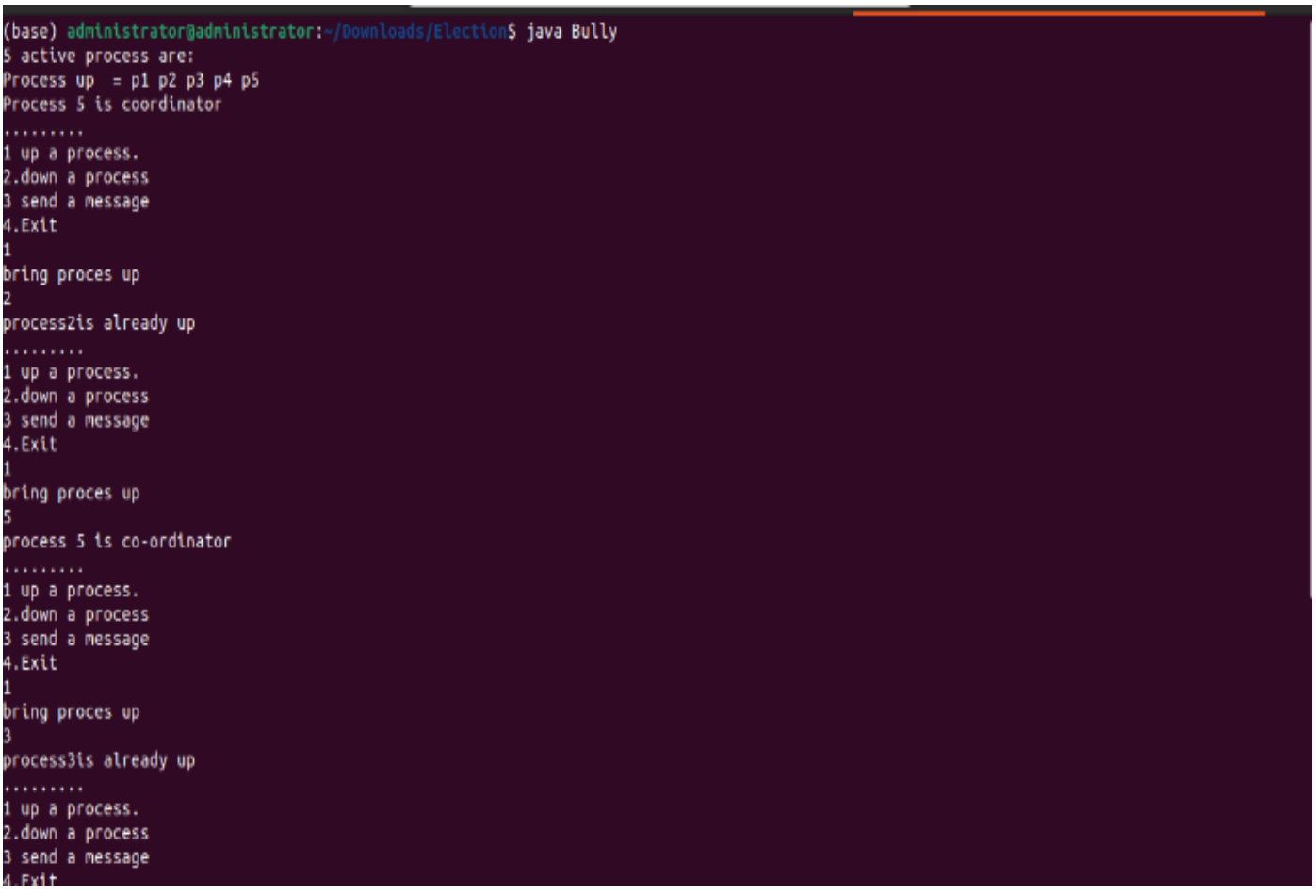
```
package ass6;
import java.util.Scanner;

public class Ring {
        public static void main(String[] args) {
                int temp, i, j;
                char str[] = new char[10];
                Rr proc[] = new Rr[10];
                for (i = 0; i < proc.length; i++)
                        proc[i] = new Rr();
                Scanner in = new Scanner(System.in);
                System.out.println("Enter the number of process : ");
                int num = in.nextInt();
                for (i = 0; i < num; i++) {
                        proc[i].index = i;
                        System.out.println("Enter the id of process : ");
                        proc[i].id = in.nextInt();
                        proc[i].state = "active";
                        proc[i].f = 0;
                }
                for (i = 0; i < num - 1; i++) {
                        for (j = 0; j < num - 1; j++) {
                                if (proc[j].id > proc[j + 1].id) {
                                        temp = proc[j].id;
                                        proc[j].id = proc[j + 1].id;
                                        proc[j + 1].id = temp;
                                }
                        }
                }
                for (i = 0; i < num; i++) {
                        System.out.print(" [" + i + "]" + " " + proc[i].id);
                }
                int init;
                int ch;
                int temp1;
                int temp2;
                int ch1;
                int arr[] = new int[10];
                proc[num - 1].state = "inactive";
                System.out.println("\n process " + proc[num - 1].id + "select as co-ordinator");
                while (true) {
                        System.out.println("\n 1.election 2.quit ");
                        ch = in.nextInt();
                        for (i = 0; i < num; i++) {
                                proc[i].f = 0;
                        }
                        switch (ch) {
```

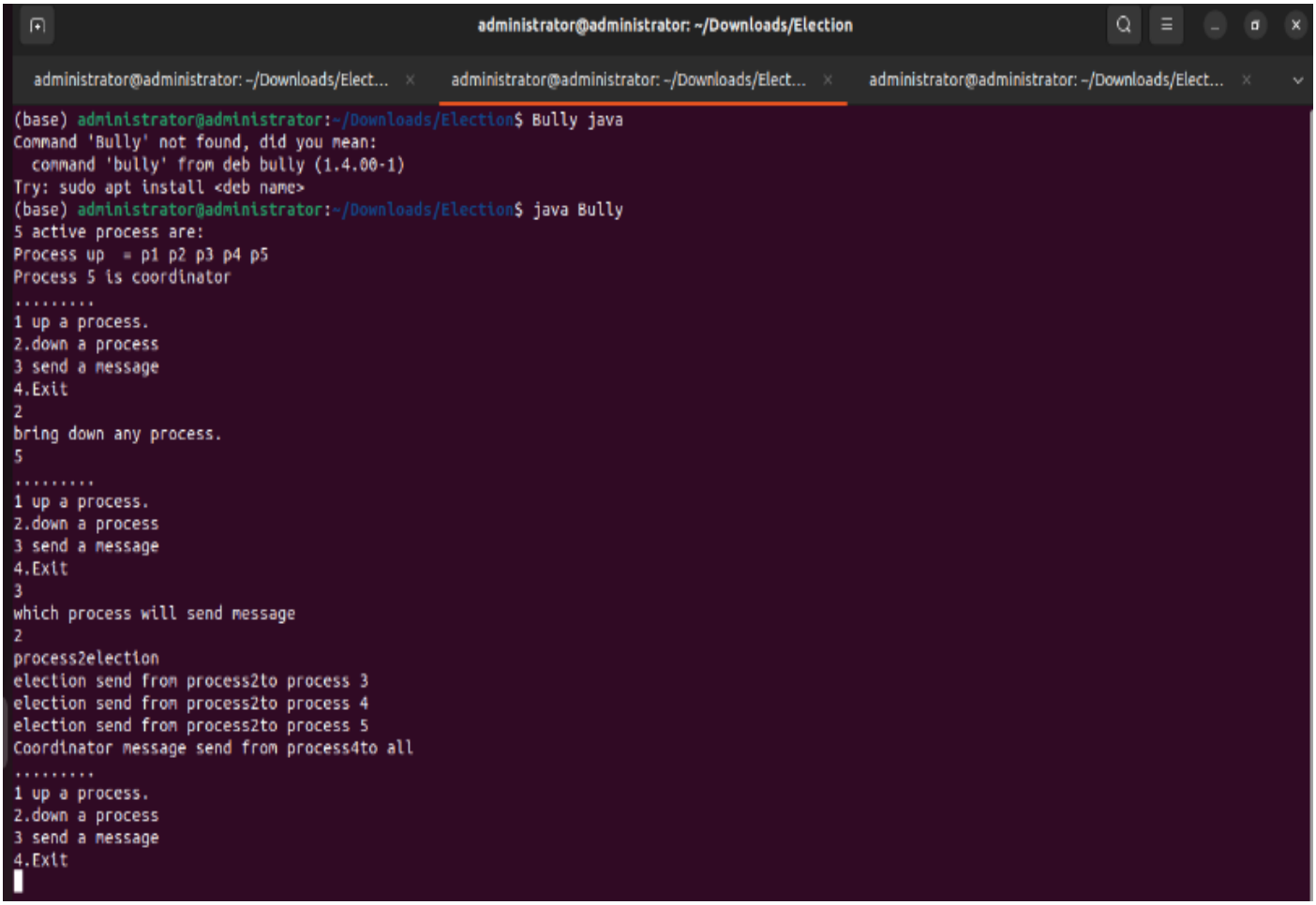
```
                                case 1:
                                        System.out.println("\n Enter the Process number who initialsied
election : ");
                                        init = in.nextInt();
                                        temp2 = init;
                                        temp1 = init + 1;
                                        i = 0;
                                        while (temp2 != temp1) {
                                                if ("active".equals(proc[temp1].state) && proc[temp1].f == 0) {
                                                        System.out.println("\nProcess " + proc[init].id + "send
message to " + proc[temp1].id);
                                                        proc[temp1].f = 1;
                                                        init = temp1;
                                                        arr[i] = proc[temp1].id;
                                                        i++;
                                                }
                                                if (temp1 == num) {
                                                        temp1 = 0;
                                                } else {
                                                        temp1++;
                                                }
                                        }
                                        System.out.println("\nProcess " + proc[init].id + " send message to " +
proc[temp1].id);
                                        arr[i] = proc[temp1].id;
                                        i++;
                                        int max = -1;
                                        for (j = 0; j < i; j++) {
                                                if (max < arr[j]) {
                                                        max = arr[j];
                                                }
                                        }
                                        System.out.println("\n process " + max + "select as co-ordinator");
                                        for (i = 0; i < num; i++) {
                                                if (proc[i].id == max) {
                                                        proc[i].state = "inactive";
                                                }
                                        }
                                        break;
                                case 2:
                                        System.out.println("Program terminated ...");
                                        return;
                                default:
                                        System.out.println("\n invalid response \n");
                                        break;
                                }
                        }
                }
        }
}

class Rr {
        public int index;
        public int id;
        public int f;
        String state;
```

}

**Outputs:**


administrator@administrator: ~/Downloads/Election
(base) administrator@administrator:~/Downloads/Election$ java Ring
Enter the number of process :
3
Enter the id of process :
5 6 8
Enter the id of process :
Enter the id of process :
[0] 5 [1] 6 [2] 8
process 8select as co-ordinator
1.election 2.quit
1
Enter the Process number who initialsied election :
2
Process 8 send message to 5
Process 5 send message to 6
Process 6 send message to 8
process 8select as co-ordinator


(base) administrator@administrator:~/Downloads/Election$ java Bully
5 active process are:
Process up = p1 p2 p3 p4 p5
Process 5 is coordinator
.....
1 up a process.
2.down a process
3 send a message
4.Exit
1
bring proces up
2
process2is already up
.....
1 up a process.
2.down a process
3 send a message
4.Exit
1
bring proces up
5
process 5 is co-ordinator
.....
1 up a process.
2.down a process
3 send a message
4.Exit
1
bring proces up
3
process3is already up
.....
1 up a process.
2.down a process
3 send a message
4.Exit


administrator@administrator: ~/Downloads/Election
administrator@administrator: ~/Downloads/Elect...
administrator@administrator: ~/Downloads/Elect...
administrator@administrator: ~/Downloads/Elect...
(base) administrator@administrator:~/Downloads/Election$ Bully java
Command 'Bully' not found, did you mean:
command 'bully' from deb bully (1.4.00-1)
Try: sudo apt install <deb name>
(base) administrator@administrator:~/Downloads/Election$ java Bully
5 active process are:
Process up = p1 p2 p3 p4 p5
Process 5 is coordinator
.....
1 up a process.
2.down a process
3 send a message
4.Exit
2
bring down any process.
5
.....
1 up a process.
2.down a process
3 send a message
4.Exit
3
which process will send message
2
process2election
election send from process2to process 3
election send from process2to process 4
election send from process2to process 5
Coordinator message send from process4to all
.....
1 up a process.
2.down a process
3 send a message
4.Exit