

Assignment No. 6

Name: Viraj Sarjerao Kawade

Roll No: 043

BE-(IT)

Subject: LP-V (Distributed System)

Code: Bully Algorithm

```
1. from statistics import mode
2.
3. class Process:
4.     def __init__(self, process_id, total_count):
5.         self.process_id = process_id
6.         self.total_count = total_count
7.         self.leader_id = -1
8.         self.is_active = True
9.
10.    def crash(self):
11.        self.is_active = False
12.
13.    def start(self):
14.        self.is_active = True
15.
16.    def is_leader(self):
17.        if self.process_id == self.leader_id:
18.            return True
19.        return False
20.
21.    def set_leader(self, leader):
22.        self.leader_id = leader
23.
24.    def get_leader(self):
25.        return self.leader_id
26.
27.    def sendRequest(self, toProcess):
28.        print(f"Sending request to process {toProcess.process_id} from {self.process_id}")
29.        if toProcess.receiveRequest(self.process_id):
30.            print(f"Ok recived from {toProcess.process_id}")
31.            self.set_leader(toProcess.process_id)
32.        else:
33.            print(f"No response from {toProcess.process_id}")
34.
35.
36.    def receiveRequest(self, fromProcess):
37.        if(self.is_active):
38.            print(f"Recived request from process {fromProcess}.")
39.            return self.recivedMessage()
40.        return False
41.
42.    def recivedMessage(self):
43.        return True;
44.
45. class Bully:
46.     def __init__(self, total_count):
47.         self.processes = []
48.         self.total_count = total_count
49.         # self.leader = None
50.
51.     def intiaialzeProcesses(self):
52.         self.processes = []
53.         for i in range(self.total_count):
54.             self.processes.append(Process(i, total_count = self.total_count))
55.         self.elect_leader()
56.         self.coordinator()
57.
58.     def elect_leader(self, current=0):
59.         for i in range(current, self.total_count):
60.             if self.processes[i].is_active:
61.                 # [self.processes[i].sendRequest(self.processes[j]) for j in range(i,
self.total_count)]
```

```

62.         for j in range(i+1, self.total_count):
63.             if(self.processes[j].is_active):
64.                 self.processes[i].sendRequest(self.processes[j])
65.             elif(not self.processes[j].is_active and i+1==self.total_count-1):
66.                 self.processes[i].sendRequest(self.processes[i])
67.
68.             if self.processes[i].get_leader()==-1:
69.                 self.processes[i].sendRequest(self.processes[i])
70.             # if(i==self.total_count-1):
71.             #     self.processes[i].sendRequest(self.processes[i])
72.
73.     def crash(self, crash_id):
74.         if(crash_id<self.total_count and crash_id>=0):
75.             self.processes[crash_id].crash()
76.             # print(f"Process id {Process.process_id} crashed.")
77.             if(self.processes[crash_id].is_leader()):
78.                 print("Leader process Down.\n Initialing the leader lookout.")
79.                 self.elect_leader(0)
80.
81.     def start(self, process_id):
82.         if(self.processes[process_id].is_active):
83.             print("Process already active")
84.         else:
85.             self.processes[process_id].start()
86.             self.elect_leader()
87.             # if(self.processes[process_id].is_active):
88.             #     if process_id>self.processes[self.leader].get_leader():
89.             #         self.elect_leader(self.leader)
90.
91.     def coordinator(self):
92.         leader = []
93.         for p in self.processes:
94.
95.             if p.is_active:
96.                 print(p.get_leader())
97.                 leader.append(p.get_leader())
98.
99.         self.leader = mode(leader)
100.

```

Driver.py

```

1. from Bully import Bully
2. #Dummy Processes
3.
4. process_count = int(input("Enter Number of Processes"))
5. bully = Bully(process_count)
6. bully.intiailzeProcesses()
7.
8. state = True
9.
10. while state:
11.     print("1. Initialize the process\n2. Bring Down process\n3. Activate Process\n4. Exit \n
12.     choice = int(input())
13.     if(choice==1):
14.         bully.intiailzeProcesses()
15.
16.     elif(choice==2):
17.         crash_id = int(input("Enter the process you want to crash"))
18.         bully.crash(crash_id)
19.
20.     elif(choice==3):
21.         process_id = int(input("Enter the process you want to start"))
22.         bully.start(process_id)

```

```

23.
24.     elif(choice==4):
25.         state=False
26.         print("Exiting the program")
27.
28.     elif(choice==5):
29.         bully.coordinator()
30.     else:
31.         print("Invalid Input")
32.

```

Ring Algorithm

```

1. class Pro:
2.     def __init__(self, id):
3.         self.id = id
4.         self.act = True
5.
6. class GFG:
7.     def __init__(self):
8.         self.TotalProcess = 0
9.         self.process = []
10.
11.     def initialiseGFG(self):
12.         print("No of processes 5")
13.         self.TotalProcess = 5
14.         self.process = [Pro(i) for i in range(self.TotalProcess)]
15.
16.     def Election(self):
17.         print("Process no " + str(self.process[self.FetchMaximum()].id) + " fails")
18.         self.process[self.FetchMaximum()].act = False
19.         print("Election Initiated by 2")
20.         initializedProcess = 2
21.
22.         old = initializedProcess
23.         newer = old + 1
24.
25.         while (True):
26.             if (self.process[newer].act):
27.                 print("Process " + str(self.process[old].id) + " pass
Election(" + str(self.process[old].id) + ") to" + str(self.process[newer].id))
28.                 old = newer
29.                 newer = (newer + 1) % self.TotalProcess
30.                 if (newer == initializedProcess):
31.                     break
32.32.
33.         print("Process " + str(self.process[self.FetchMaximum()].id) + " becomes
coordinator")
34.         coord = self.process[self.FetchMaximum()].id
35.
36.         old = coord
37.         newer = (old + 1) % self.TotalProcess
38.         while (True):
39.             if (self.process[newer].act):
40.                 print("Process " + str(self.process[old].id) + " pass
Coordinator(" + str(coord) + ") message to process " + str(self.process[newer].id))
41.                 old = newer
42.                 newer = (newer + 1) % self.TotalProcess
43.                 if (newer == coord):
44.                     print("End Of Election ")
45.                     break
46.46.
47.     def FetchMaximum(self):
48.         maxId = -9999
49.         ind = 0
50.         for i in range(self.TotalProcess):
51.             if (self.process[i].act and self.process[i].id > maxId):

```

```

52.                                     maxId = self.process[i].id
53.                                     ind = i
54.                                     return ind
55.
56. def main():
57.     object = GFG()
58.     object.initialiseGFG()
59.     object.Election()
60.
61. if __name__ == "__main__":
62.     main()
63.

```

Output:

After Implementing The Bully Algorithm:

```

ca Select Command Prompt - python Driver.py
Microsoft Windows [Version 10.0.19045.2965]
(c) Microsoft Corporation. All rights reserved.

C:\Users\hp>cd C:\Users\hp\Desktop\ds lab\Assignment No.6

C:\Users\hp\Desktop\ds lab\Assignment No.6>python Bully.py

C:\Users\hp\Desktop\ds lab\Assignment No.6>python Driver.py
Enter the number of processes: 5
Sending request to process 1 from 0
Received request from process 0.
Ok received from 1
Sending request to process 2 from 0
Received request from process 0.
Ok received from 2
Sending request to process 3 from 0
Received request from process 0.
Ok received from 3
Sending request to process 4 from 0
Received request from process 0.
Ok received from 4
Sending request to process 2 from 1
Received request from process 1.
Ok received from 2
Sending request to process 3 from 1
Received request from process 1.
Ok received from 3
Sending request to process 4 from 1
Received request from process 1.
Ok received from 4
Sending request to process 3 from 2
Received request from process 2.
Ok received from 3
Sending request to process 4 from 2
Received request from process 2.
Ok received from 4
Sending request to process 4 from 3
Received request from process 3.
Ok received from 4
Sending request to process 4 from 4
Received request from process 4.
Ok received from 4
4
4
4
4
4
1. Initialize the process
2. Bring down process
3. Activate process

```

After Implementing The Ring Algorithm:

Command Prompt

Exiting the program

C:\Users\hp\Desktop\ds lab\Assignment No.6>python Ring.py

No of processes: 5

Process no 4 fails

Election Initiated by 2

Process 2 pass Election(2) to 3

Process 3 pass Election(3) to 0

Process 0 pass Election(0) to 1

Process 3 becomes coordinator

Process 3 pass Coordinator(3) message to process 0

Process 0 pass Coordinator(3) message to process 1

Process 1 pass Coordinator(3) message to process 2

End Of Election

C:\Users\hp\Desktop\ds lab\Assignment No.6>_