IN4150 exercise 3c
Group 12

Code repository: https://github.com/xjmxmt/da_exercises/tree/main/exercise_three/src

Test cases:
Test case 1: only one candidate process (Process 10) is trying to get elected,
Log:
Process 10[8, 7, 9, 1, 2, 4, 5, 6, 3]
10 send to 8
8 receive (0, 10) from 10
Process 8 set 10 as parent and ack
10 receive (0, 10) from 8
10 send to 7
7 receive (1, 10) from 10
Process 7 set 10 as parent and ack
10 receive (1, 10) from 7
10 send to 9
9 receive (2, 10) from 10
Process 9 set 10 as parent and ack
10 receive (2, 10) from 9
10 send to 1
1 receive (3, 10) from 10
Process 1 set 10 as parent and ack
10 receive (3, 10) from 1
10 send to 2
2 receive (4, 10) from 10
Process 2 set 10 as parent and ack
10 receive (4, 10) from 2
10 send to 4
4 receive (5, 10) from 10
Process 4 set 10 as parent and ack
10 receive (5, 10) from 4
10 send to 5
5 receive (6, 10) from 10
Process 5 set 10 as parent and ack
10 receive (6, 10) from 5
10 send to 6
6 receive (7, 10) from 10
Process 6 set 10 as parent and ack
10 receive (7, 10) from 6
10 send to 3
3 receive (8, 10) from 10
Process 3 set 10 as parent and ack

10 receive (8, 10) from 3
Process 10 ELECTED

Result:

Process id	capture msg	kill msg	ack msg	maximum level	captured time
1 to 9	Receive 1	0	Send 1	0	1
10	Send 9	0	Receive 9	9	0

For ordinary processes with id from 1 to 9, the results are the same. For Process 10, it will capture every other node in a random order.

Test case 2: two candidate processes (Process 9, 10) start the election simultaneously,
Log:
Process 9[5, 6, 3, 10, 4, 1, 2, 7, 8]
9 send to 5
Process 10[3, 8, 7, 5, 9, 2, 1, 6, 4]
10 send to 3
5 receive (0, 9) from 9
Process 5 set 9 as parent and ack
9 receive (0, 9) from 5
9 send to 6
3 receive (0, 10) from 10
Process 3 set 10 as parent and ack
6 receive (1, 9) from 9
Process 6 set 9 as parent and ack
10 receive (0, 10) from 3
10 send to 8
9 receive (1, 9) from 6
9 send to 3
3 receive (2, 9) from 9
Process 3 attempt to kill 10
10 receive (2, 9) from 3
Process 10 RESIGNED
3 receive (2, 9) from 10
Process 3 set 9 as parent and ack
9 receive (2, 9) from 3
9 send to 10
8 receive (1, 10) from 10
Process 8 set 10 as parent and ack
10 receive (1, 10) from 8
Process 10 attempt to kill 9
9 receive (1, 10) from 10
10 receive (3, 9) from 9
Process 10 attempt to kill 9
9 receive (3, 9) from 10

9 send to 4
4 receive (4, 9) from 9
Process 4 set 9 as parent and ack
9 receive (4, 9) from 4
9 send to 1
1 receive (5, 9) from 9
Process 1 set 9 as parent and ack
9 receive (5, 9) from 1
9 send to 2
2 receive (6, 9) from 9
Process 2 set 9 as parent and ack
9 receive (6, 9) from 2
9 send to 7
7 receive (7, 9) from 9
Process 7 set 9 as parent and ack
9 receive (7, 9) from 7
9 send to 8
8 receive (8, 9) from 9
Process 8 attempt to kill 10
10 receive (8, 9) from 8
Process 10 attempt to kill 9
9 receive (8, 9) from 10
Process 9 ELECTED

Result:

Process id	capture msg	kill msg	ack msg	maximum level	captured time
1 to 8	Receive 1 or 2	0 or 1	0	0	1 or 2
9	Send 9	0	Receive 9	9	0
10	Receive 1	3	Receive 2	0	1

The log shows that Process 9 sent a message to Process 3, since Process 10 is the parent of Process 3 and the level of Process 9 was higher, Process 10 got captured and turned into an ordinary process. If Process 10 captures Process 9 first, then the result of Process 9 and Process 10 will exchange.

Test case 3: several processes (Process 7, 8, 9, 10) start the election consecutively, with a delay interval of 100 ms,
Log:
Process 7[10, 6, 5, 1, 2, 3, 8, 4, 9]
7 send to 10
10 receive (0, 7) from 7
Process 8[6, 5, 10, 2, 7, 4, 1, 9, 3]
8 send to 6
6 receive (0, 8) from 8
Process 6 set 8 as parent and ack
8 receive (0, 8) from 6

8 send to 5
Process 9[10, 1, 5, 2, 3, 8, 7, 6, 4]
9 send to 10
5 receive (1, 8) from 8
Process 5 set 8 as parent and ack
8 receive (1, 8) from 5
8 send to 10
Process 10[3, 5, 2, 1, 8, 6, 9, 4, 7]
10 send to 3
10 receive (0, 9) from 9
10 receive (2, 8) from 8
Process 10 RESIGNED
8 receive (2, 8) from 10
8 send to 2
2 receive (3, 8) from 8
Process 2 set 8 as parent and ack
8 receive (3, 8) from 2
8 send to 7
3 receive (0, 10) from 10
Process 3 set 10 as parent and ack
10 receive (0, 10) from 3
Process 10 attempt to kill 8
8 receive (0, 10) from 10
7 receive (4, 8) from 8
Process 7 RESIGNED
8 receive (4, 8) from 7
8 send to 4
4 receive (5, 8) from 8
Process 4 set 8 as parent and ack
8 receive (5, 8) from 4
8 send to 1
1 receive (6, 8) from 8
Process 1 set 8 as parent and ack
8 receive (6, 8) from 1
8 send to 9
9 receive (7, 8) from 8
Process 9 RESIGNED
8 receive (7, 8) from 9
8 send to 3
3 receive (8, 8) from 8
Process 3 attempt to kill 10
10 receive (8, 8) from 3
Process 10 attempt to kill 8
8 receive (8, 8) from 10

Process 8 ELECTED

Result:

Process id	capture msg	kill msg	ack msg	maximum level	captured time
1 to 6	Receive 1 or 2	0 or 1	0	0	1 or 2
7	Receive 1	0	0	0	1
8	Send 9	0	Receive 9	9	0
9	Receive 1	0	0	0	1
10	Receive 1, Send 1	2	Receive 1	1	1

There are many other possible results, and only this one is shown because Process 8 first captures all the other candidate processes.

Test case 4: all five processes start the election simultaneously,

Log:

Process 2[3, 1, 4, 5]
2 send to 3
Process 5[4, 3, 1, 2]
5 send to 4
3 receive (0, 2) from 2
Process 4[1, 2, 3, 5]
4 send to 1
Process 3[5, 1, 2, 4]
3 send to 5
Process 1[2, 4, 5, 3]
1 send to 2
4 receive (0, 5) from 5
Process 4 RESIGNED
5 receive (0, 5) from 4
5 send to 3
3 receive (1, 5) from 5
Process 3 RESIGNED
5 receive (1, 5) from 3
5 send to 1
5 receive (0, 3) from 3
1 receive (0, 4) from 4
Process 1 RESIGNED
4 receive (0, 4) from 1
Process 4 attempt to kill 5
5 receive (0, 4) from 4
1 receive (2, 5) from 5
Process 1 attempt to kill 4
4 receive (2, 5) from 1
Process 4 attempt to kill 5
5 receive (2, 5) from 4

5 send to 2
2 receive (0, 1) from 1
2 receive (3, 5) from 5
Process 2 RESIGNED
5 receive (3, 5) from 2
Process 5 ELECTED

Result:

Process id	capture msg	kill msg	ack msg	maximum level	captured time
1	Receive 1	1	0	0	1
2	Receive 1	0	0	0	1
3	Receive 1	0	0	0	1
4	Receive 1	2	0	0	1
5	Send 4	0	Receive 4	4	0

There are many other possible outcomes, and only this one is shown because all the other candidate processes were captured except for Process 5.