

DISTRIBUTED ALGORITHMS

IMPLEMENTATION OF ELECTION IN ASYNCHRONOUS COMPLETE NETWORKS

by

João Almeida
Manuel Ribeiro

TESTS

To guarantee the quality of our implementation we ran multiple tests, a summary of those tests is present in Table 1. To ensure the algorithm was implemented correctly, information regarding the state of the processes was printed during the execution of the algorithm. All these test are explained in the section *Detailed Tests*.

For each test we measured a series of parameters to evaluate the output of the algorithm. The parameters are:

- **Number of Level Increases:** Sum of the *Levels* of all processes.
- **Number of Captures:** Number of *Captures* that were successful, measured by the ordinary processes.
- **Number of Kill messages:** Number of messages sent by ordinary processes to *Kill* their previous owner.
- **Number of Acknowledgment messages:** Number of *Acknowledgment* messages sent for each *kill* or *Capture* messages.
- **Number of Capture/Level Discrepancies:** The difference between the number of *Captures* and number of *Level Increases*.

Test #	# Machines	Total # Processes	#Level increases	# Captures	# Kills	# Acks	#Capture/Level Discrepancies
1	1	2	1	1	0	1	0
2	1	5	8	8	3	11	0
3	2	10	17	17	7	24	0
4	1	10	16	16	9	25	0
5	1	20	35	35	16	51	0
6	1	50	106	110	63	169	4
7	1	100	213	221	125	338	8
8	1	250	543	565	322	865	22
9	1	500	1128	1180	690	1818	52
10	1	750	1712	1794	1063	2775	82

Table 1: Tests summary

As you can see there is a difference between the number of *Captures* and the *Level Increases*, for the larger tests. Theoretically these should be the same, however due to a known problem of synchronization some of these captures do not result in a level increase for the candidate because it is killed before receiving the acknowledgment. This acknowledgment arrives later, however it is ignored, as the candidate is already dead. So this way the number of acknowledgments respects: $\#ACKS = \#KILLS + \#CAPTURES - \#DISCREPANCIES$.

DETAILED TESTS

```

Test 1 output
[INFO] Connections Ready
[Process: 1] [C] Is now a Candidate.
[Process: 2] [O] Captured by Candidate Process: 1.
[Process: 1] [C] Elected!!
[Process: 1] [C] Level = 2. Times Captured = 0. Acks = 1.
[Process: 2] [C] Level = 1. Times Captured = 1. Acks = 0.
[INFO]
Level Sum - Number of Processes = 1.
Captures Sum = 1
Total Kills = 0
Total Acks = 1
Missed captures: 0

```

In this example, *Test 1*, two processes are trying to get elected. Process 1 is the first candidate and captures Process 2 immediately, making itself the elected process. The '*Level Sum - Number of Processes*' is equal to the '*Captures Sum*', which proves the correctness of the algorithm for two processes.

```

Test 4 output
[INFO] Connections Ready
[Process: 9] [C] Is now a Candidate.
[Process: 1] [C] Is now a Candidate.
[Process: 7] [C] Is now a Candidate.
[Process: 5] [C] Is now a Candidate.
[Process: 4] [C] Is now a Candidate.
[Process: 2] [C] Is now a Candidate.
[Process: 8] [C] Is now a Candidate.
[Process: 3] [C] Is now a Candidate.
[Process: 10] [C] Is now a Candidate.
[Process: 6] [C] Is now a Candidate.
[Process: 5] [O] Captured by Candidate Process: 4.
[Process: 8] [O] Captured by Candidate Process: 1.
[Process: 9] [O] Captured by Candidate Process: 2.
[Process: 2] [O] Captured by Candidate Process: 5.
[Process: 4] [O] Captured by Candidate Process: 9.
[Process: 6] [O] Captured by Candidate Process: 7.
[Process: 3] [O] Captured by Candidate Process: 4.
[Process: 10] [O] Captured by Candidate Process: 7.
[Process: 1] [O] Captured by Candidate Process: 2.
[Process: 1] [C] Was Killed
[Process: 1] [C] Received a kill Message, but was already killed.
[Process: 8] [O] Captured by Candidate Process: 7.
[Process: 1] [C] Received a kill Message, but was already killed.
[Process: 1] [O] Captured by Candidate Process: 7.
[Process: 2] [C] Was Killed
[Process: 5] [O] Captured by Candidate Process: 7.
[Process: 4] [C] Was Killed
[Process: 2] [O] Captured by Candidate Process: 7.
[Process: 5] [C] Was Killed
[Process: 9] [O] Captured by Candidate Process: 7.
[Process: 2] [C] Received a kill Message, but was already killed.
[Process: 4] [O] Captured by Candidate Process: 7.
[Process: 9] [C] Was Killed
[Process: 3] [O] Captured by Candidate Process: 7.

```

```
[Process: 4]      [C]      Received a kill Message, but was already killed.
[Process: 7]      [C]      Elected!!
[Process: 1]      [C]      Level = 2. Times Captured = 2. Acks = 4.
[Process: 2]      [C]      Level = 3. Times Captured = 2. Acks = 4.
[Process: 3]      [C]      Level = 1. Times Captured = 2. Acks = 0.
[Process: 4]      [C]      Level = 3. Times Captured = 2. Acks = 4.
[Process: 5]      [C]      Level = 2. Times Captured = 2. Acks = 2.
[Process: 6]      [C]      Level = 1. Times Captured = 1. Acks = 0.
[Process: 7]      [C]      Level = 10. Times Captured = 0. Acks = 9.
[Process: 8]      [C]      Level = 1. Times Captured = 2. Acks = 0.
[Process: 9]      [C]      Level = 2. Times Captured = 2. Acks = 2.
[Process: 10]     [C]      Level = 1. Times Captured = 1. Acks = 0.
[INFO]
Level Sum - Number of Processes = 16
Captures Sum = 16
Total Kills = 9
Total Acks = 25
Missed captures: 0
```