Jesús Castillo Benito
1/11/2024

03 - Mutual exclusion algorithms

a)

Requirements for mutual exclusion:

- Mutual exclusion: only one thread can be in the critical section at any given time.
- Progress: if there is no thread in the critical section, the next thread to be in the critical section must depend only on the threads involved in the contention
- Bounded waiting: each thread should have a bounded number of times it needs to wait before entering the critical section, ensuring no thread is indefinitely postponed.

Knowing the anatomy of Peterson's algorithm,

Proof of correctness:

1. Mutual exclusion
   For both threads to enter their critical sections, they must have each set their respective in1 and in2 to 1.
   Then, since both threads would have also set last to point to the other, one thread would have been forced to wait by checking the while loop condition. This ensures that only one of the threads can proceed.

2. Progress
   If only one thread want to enter the critical section, it will not wait, ensuring progress in our program. If both of them try to access the critical section at the same time, only the one for whom last points into the other will wait. Meanwhile the other thread will enter its critical section.

3. Bounded waiting
   The time is bound for the time that the other thread needs in the critical section. I
   If a thread is waiting, the only possible reason is that the flag from the other thread is set and last is not in its favor. But as soon as the other thread leaves the critical section, it allows the other threads to stop being blocked.

c)

During the tests of this part, I was getting strange results. The algorithm was correctly implemented but c1 and c2 weren’t getting updated, so in all the tests I was getting all the time c1 + c2 = 0.

In this test, the only thing that changed was changing was the execution time, that increased respectively when the iterations were growing and from 2^{15} (32768), the result of s stops being 0, to start being the number of iterations of the program.

The system that I am using has 8 CPUs according to the “top” monitor and a maximum value of signed integer of 2147483647.