

READERS AND WRITERS

THE RISE OF THE WRITERS

Alessandro Monticelli, Jóhannes Helgi Tómasson, Jonas Stahl

April 3, 2026

1 The story of all the semaphores

The semaphores m_1 and m_2 are pretty straight forward. First one is to avoid collisions when writing to the variable n_r and the other is for n_w . Without them we cannot guarantee sane results for either of them. But by this we can now guarantee that both variables are edited correctly and assure correct counting.

We now need to assure that every writer has exclusive access. The semaphore responsible for this is $writer$. There are three situations that could appear:

- no one has access, here the semaphore is not set
- one writer has access, here the currently active writer holds the semaphore
- multiple readers have access, here the first reader acquires and the last one releases the semaphore

A writer can only get access if he acquires the semaphore. The first readers can get access only when he acquires the semaphore. All further readers can access the data without acquiring this semaphore. They are still safe as the first reader acquired it. This procedure for the readers is protected by the semaphore m_1 . As a result further readers cannot get past the first reader and we guarantee that every reader checks the number of readers in a sequential order. Also the releasing is protected by m_1 to guarantee correct behaviour.

By now we have a correct behaviour overall, but without the priority for writers. This priority is guaranteed by the semaphore $reader$. Every reader needs to request this semaphore before it is able to get read access. If this semaphore would be blocked, the number of active readers can only decrease till none is left. This leads to the following needed assumption: As long as a writer is writing or waiting to write, the $reader$ semaphore should be set to prevent new readers. To achieve this we track the number of currently interested writers in n_w . As the variable is protected as described in the beginning, the first appearing writer can lock the $reader$ semaphore. Every writer coming later only increases the number of writers n_w . The semaphore $reader$ is only released after the last writer is finished (when n_w is 0). This prevents readers from slipping between writers.

The most critical part is where the first writer tries to lock the semaphore $reader$. Without the semaphore m_3 it could happen, that the writer needs to compete against multiple readers. With the semaphore m_3 we can guarantee that only one other reader is able to compete against the writer. As we have no FIFO or other fairness guarantee about semaphore this gives us a

better chance for the writer to succeed. But if we have no strong semaphore guarantees it could still happen that the writer waits forever. Usually this is very unlikely, so the writer should have preference.