PUTTA JAIDEEP

OPERATING SYSTEMS (CSN-232)

STARVE FREE READERS WRITERS PROBLEM

In all of the threads some threads will try to read and some threads will try to write . But as we know about the constraint that no thread can either read or write in the resource while another thread is writing to it. So, to solve this problem there are three types of solutions.

1. First readers-writers problem.
2. Second readers-writers problem.
3. Starve free readers-writers problem.

FIRST READERS-WRITERS PROBLEM:

In this the first reader will lock the resource for reading. So as the resource is locked the writer cannot have access to it hence it can't modify the resource while other files are reading it.
Once the first reader is in the entry section, it will lock the resource. Hence this will prevent any writers from accessing it. Subsequent readers can just utilize the locked resource. The reader to finish last must unlock the resource, thus making it available to writers.
Hence in this solution the stream of readers will lock out the resource from the writers which will result in the starvation of the writers.

SECOND READERS-WRITERS PROBLEM:

In this solution the constraint is that no writer, once added to the queue, shall be kept waiting longer than absolutely necessary . This solution is also called writers-preference.

Here we force every reader to lock and release the readtry semaphore individually. The writers on the other hand don't need to lock it individually. Only the first writer will lock the readtry and then all subsequent writers can simply use the resource as it gets freed by the previous writer. The very last writer must release the readtry semaphore, thus opening the gate for readers to try reading.

If there are no writers wishing to get to the resource, as indicated to the reader by the status of the readtry semaphore, then the readers will not lock the resource. This is done to allow a writer to immediately take control over the resource as soon as the current reader is finished reading. Otherwise, the writer would need to wait for a queue of readers to be done before the last one can unlock the readtry semaphore. As soon as a writer shows up, it will try to set the readtry semaphore and hang up there waiting for the current reader to release the readtry. It will then take control over the resource as soon as the current reader is done reading and lock all future readers out.

Hence this causes the readers to starve.

STARVE FREE READERS-WRITERS PROBLEM:

Unlike the solutions above the constraint in this solution is that no thread should be allowed to starve. Which means that the operation of obtaining a lock on the shared data will always terminate in a bounded amount of time.

In this solution the semaphores preserve first-in first-out ordering when blocking and releasing threads so that whenever a writer comes it will take the control and write in the resource similarly, the reader but not simultaneously. The code for this is provided in the github link.

---------- THANK YOU ----------