

Starvation free Readers-Writers Solution

- This solution uses 2 new variables- writing & readersTurn, to make a starvation free(as well as deadlock free) solution of the problem besides the class discussed deadlock free solution.
- The pseudocode uses 2 binary semaphores – mutex1 & mutex2, to update the values of readers and waitingWriters respectively so that parallel processes updating the same variable does not cause error.

❖ Data structure used in the solution and their explanation-

1. The FIFO structure – Queue
 - We need a data structure to store processes blocked by wait function call.

❖ Functions used and their explanation-

1. acquireRead() – Whenever a process wants to read the database, this function is called and checks if there are any waiting writers in the queue. If there are no waiting writers then read operation is performed otherwise the process is added to the waiting queue.
2. acquireWrite() – Whenever a process wants to write to the database, this function is called and checks if there are any readers presently reading or any writers presently writing. If any of the 2 situation is present then the process is added to the waiting queue otherwise writing is performed on the database.
3. releaseRead() – Whenever a process completes reading, it calls this function. readers is decreased by 1 and readersTurn is set to false. If readers become 0 then notify() function is called.
4. releaseWrite() – Whenever a process completes writing, it calls this function. writing is set to false, readersTurn is set to true and notify() function is called.
5. wait() – This function puts the calling process in the waiting queue.
6. notify() – This function notifies all the processes present in the queue that a read/write process is ended, and it can again try to access the database. This function notifies the processes in FCFS order.

- The basic idea in this solution to prevent starvation is to give chance to any possible waiting reader after a write operation is performed or give chance to any possible waiting writing process after a read is performed on the database with the help of readersTurn variable. In this way we can achieve alternation between reading and writing processes and hence starvation is prevented.

Name – Piyush Arya

Enrolment no. – 21114074

Branch, Batch – CSE, O3