Readers Writers
(Semaphore MultiTasking)

- The Readers-Writers problem is another classic synchronization problem that deals with multiple processes trying to read and write to a shared resource concurrently.

- The goal is to ensure that readers can access the resource simultaneously for reading while maintaining the integrity of the data, but only one writer should be allowed to write to the resource at any given time to prevent race conditions.

- Using semaphores in a multitasking environment can help address the Readers-Writers problem efficiently.

Here's a detailed explanation of how the problem can be solved using semaphores:

١. **Define two semaphores: a readerCount semaphore and a resource semaphore.**

٢. **The readerCount semaphore is used to keep track of the number of readers currently accessing the resource. It is initialized to ١ to ensure exclusive access for writers during updates.**

٣. **The resource semaphore is used to control access to the shared resource. It is initialized to allow only one process to write at a time.**

٤. **When a reader wishes to access the resource, it first checks if it can increment the readerCount semaphore. If it's the first reader, it locks the resource semaphore to prevent writers from modifying the data.**

٥. **When a reader finishes reading, it decrements the readerCount semaphore. If it's the last reader, it releases the resource semaphore, allowing writers to access the resource.**

٦. **When a writer wants to update the resource, it waits on the resource semaphore until it becomes available. Once it acquires the semaphore, it has exclusive access to the resource.**

٧. **After the writer finishes updating the resource, it releases the resource semaphore, allowing other readers or writers to access the resource.**

By using semaphores to coordinate access to the shared resource, the Readers-Writers problem can be effectively managed in a multitasking environment. This approach helps prevent race conditions, ensures data integrity, and optimizes resource utilization for both readers and writers.

Feel free to explore more advanced techniques, such as priority mechanisms for readers or writers, to further enhance the solution to the Readers-Writers problem when implementing it in a real-world scenario.

Producer: Elham Jafari

Computer Engineering