

OS-Assignment-5: CS3523

(Implement solutions to Readers-Writers problem using Semaphores)

Name - C.Asish

Roll number – CS20BTECH11010

Design of the Program :-

- The goal of this assignment is to implement and compare two solutions using semaphores for the Reader-Writers Problem. One solution with reader preference (Readers-Writers) and one fair solution (Fair-Readers-Writers).
- The program (main thread) will read from “inp-params.txt”, which contains information regarding reader and writer threads and stores these information as global variables
- n_w writer threads and n_r reader threads are created and are joined respectively.
- Here we use reader and writer functions, which simulate readers and writers and they are implemented in accordance with whether they are reader preference or fair preference.
- We use default_random_engine to generate random numbers from exponential distribution to simulate real time critical and remainder sections.
- We print logs in to output file named “RW-log.txt”, “FairRW-log.txt”. “average time.txt” consists of the average time taken for a thread to gain entry to the Critical Section for each of the algorithms: RW and Fair-RW.

Normal Reader-Writer algorithm:-

- Two semaphores mutex and rw_mutex are used in this program to attain mutual exclusion (both are initialized to 1)
- Here mutex semaphore is used for increment of reader threads (read_count)
- Rw_mutex semaphore is used to ensure mutual exclusion between writer and reader threads in CS.
- These semaphore are used in reader and writer functions.

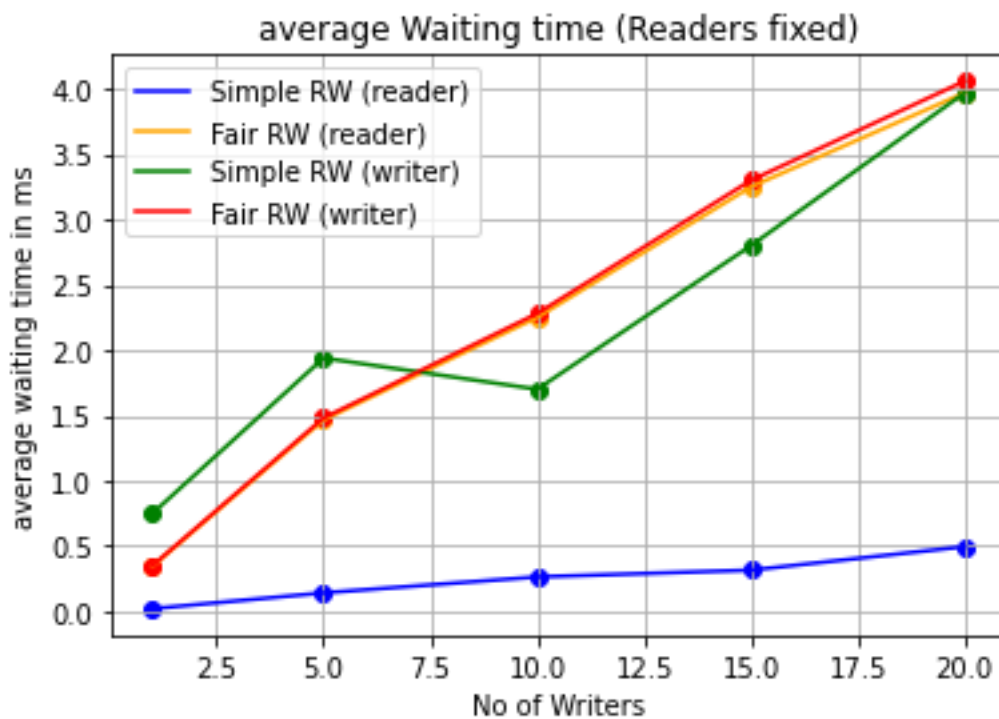
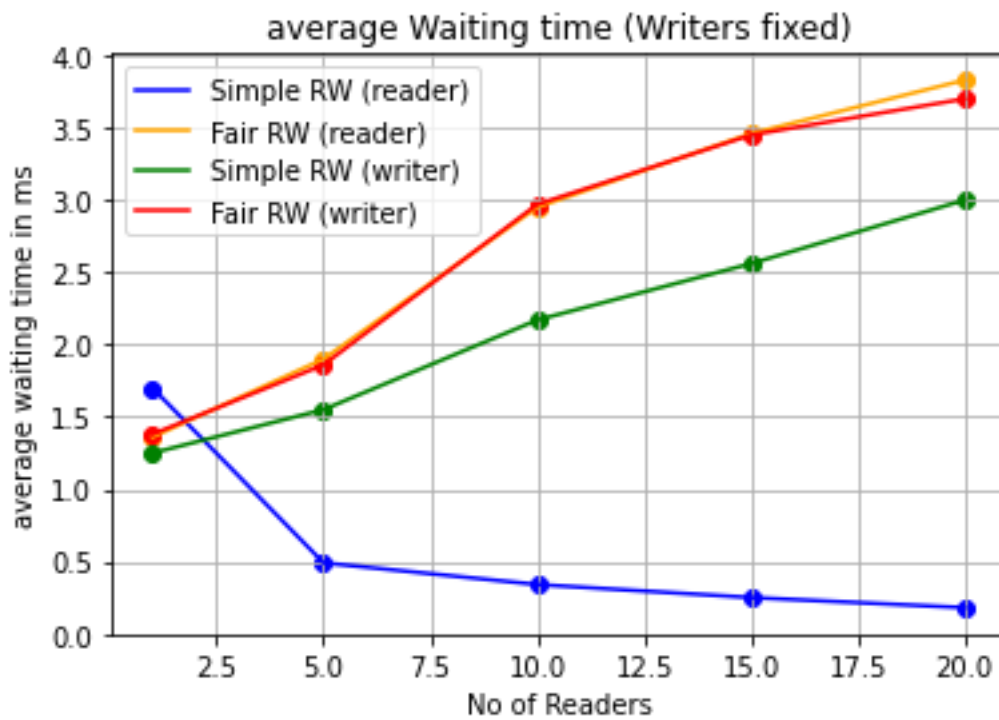
Fair-RW algorithm

- Here we use rw_mutex,mutex along with a queue (all semaphores are initialized to 1)
- Here rw_mutex and mutex work same as above algorithm and queue handles the SERVICEQUEUE of the threads(preserves order)
- These semaphore are used in reader and writer functions.

Graphs:

For all graphs, $\mu_{cs} = \mu_{rem} = 2$ and $k_r = k_w = 10$. And each graph contains 4 curves.

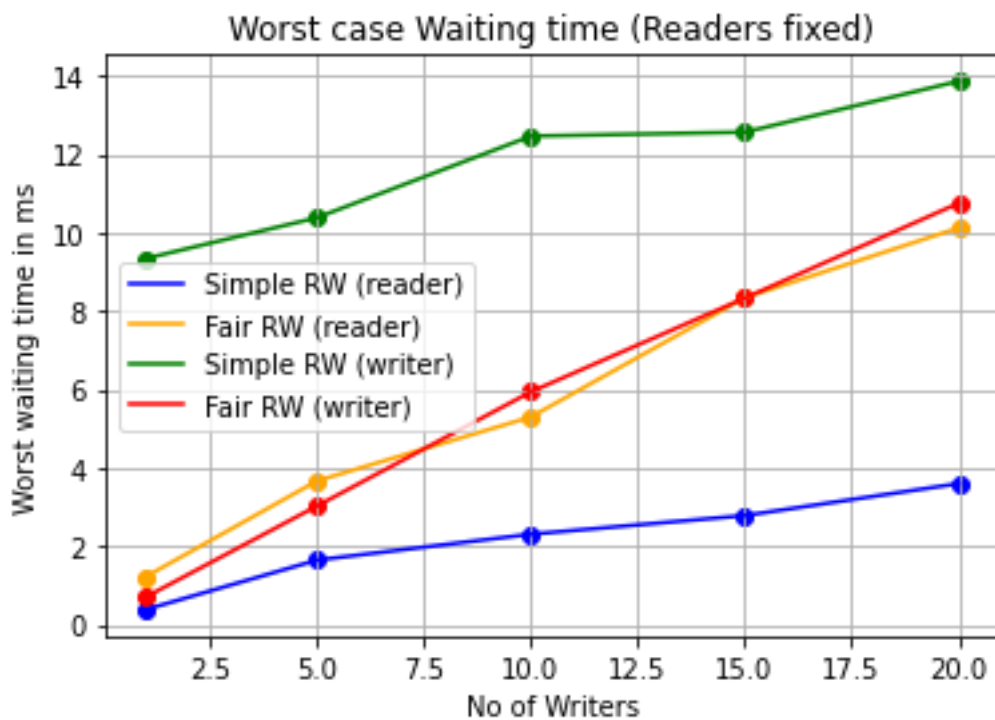
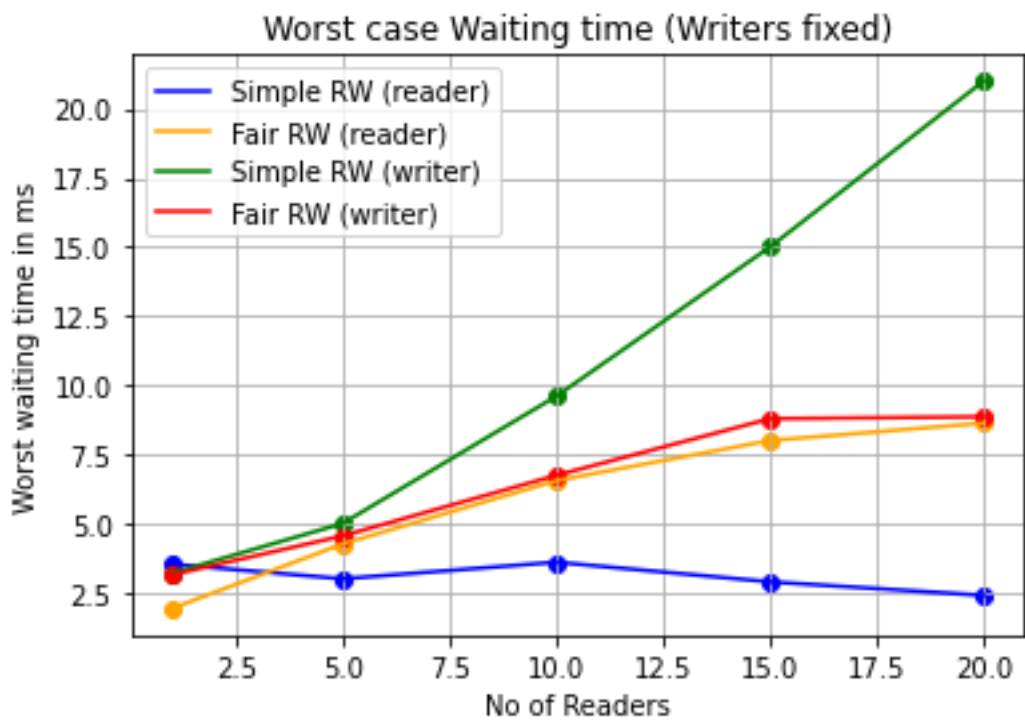
1) And 2)



From the above two graphs,

- The average waiting time of Readers threads in Normal RW algorithm is far less than waiting time of Writer threads in Normal RW algorithm .because here in Normal RW algorithm Readers are given preference over writers.
- The average waiting time of Readers threads in Normal RW algorithm is almost equal to waiting time of Writer threads in Normal RW algorithm. Because here in Fair RW algorithm Both Readers and writer threads are given equal preference.
- we can say that average waiting times of Fair RW algorithm is more than the Normal RW algorithm because to ensure fairness in Fair RW algorithm, the waiting time of threads increase.

3) and 4) graphs



From the above two graphs

- The worst case waiting time of writer threads is less in Fair RW algorithm than in Normal RW algorithm. Because in normal RW algorithm threads are writers threads may starve due to preference of reader threads more than writer threads.
- Worst case waiting time of readers is less for Normal algorithm than in Fair algorithm.
- In Fair RW algorithm worst case waiting times are almost equal and are in between reader and writer threads from normal RW algorithm.