OS PROGRAMMING ASSIGNMENT 5

Design of the program:

1. First the program will take input from inp-params.txt file and store them as global variables which represent no of reader,writer threads,their frequencies and average times in critical section,remainder section per each thread.
2. Reader threads created will start execution from reader function and writer threads will start from writer function taking argument as id of the thread(manually assigned in the program).
3. Reader and writer functions, which simulate readers and writers ,are implemented in accordance with whether they are reader preference or fair preference.
4. I used default_random_engine to generate random numbers from exponential distribution to simulate real time critical and remainder sections.
5. We print the output into files named “RW-log.txt” and “FairRW-log.txt” and “average time.txt” consists of the average waiting time of a thread to enter the critical section.

Reader writers (biased):

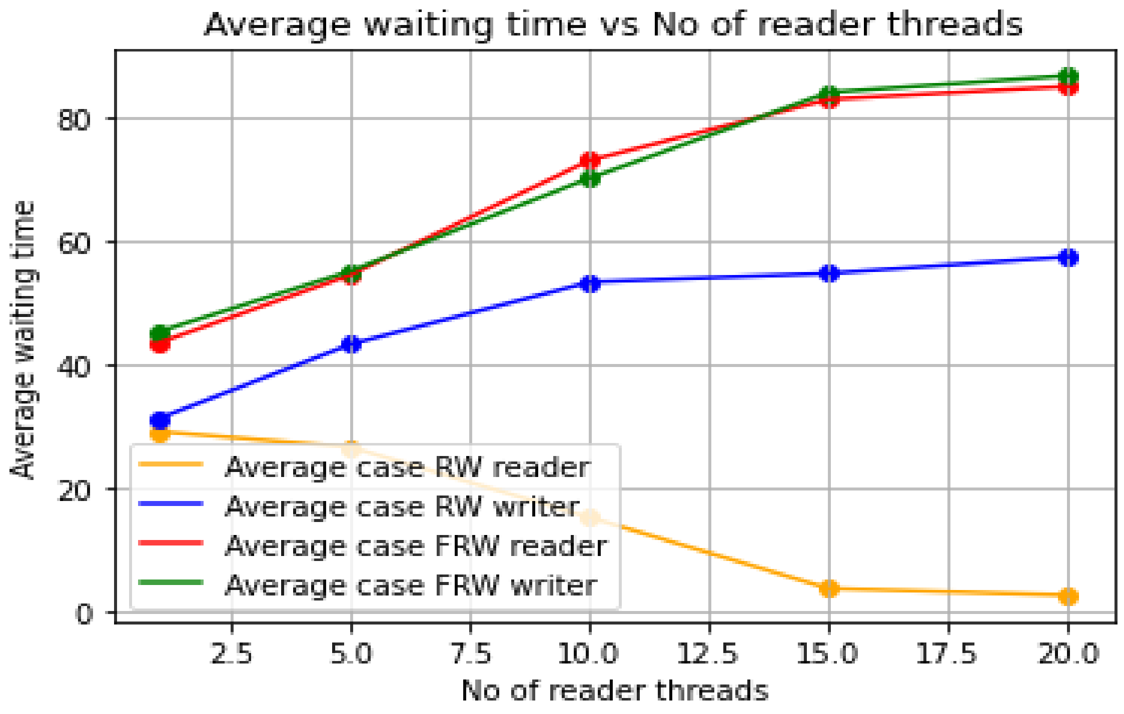
1. In this program we use two semaphores mutex and rw_mutex to attain mutual exclusion (both are initialized to 1).
2. The mutex semaphore is used for increment of reader threads(read_count) and rw_mutex semaphore is used to ensure mutual exclusion between writer and reader threads in the CS.

Fair Reader-writers:

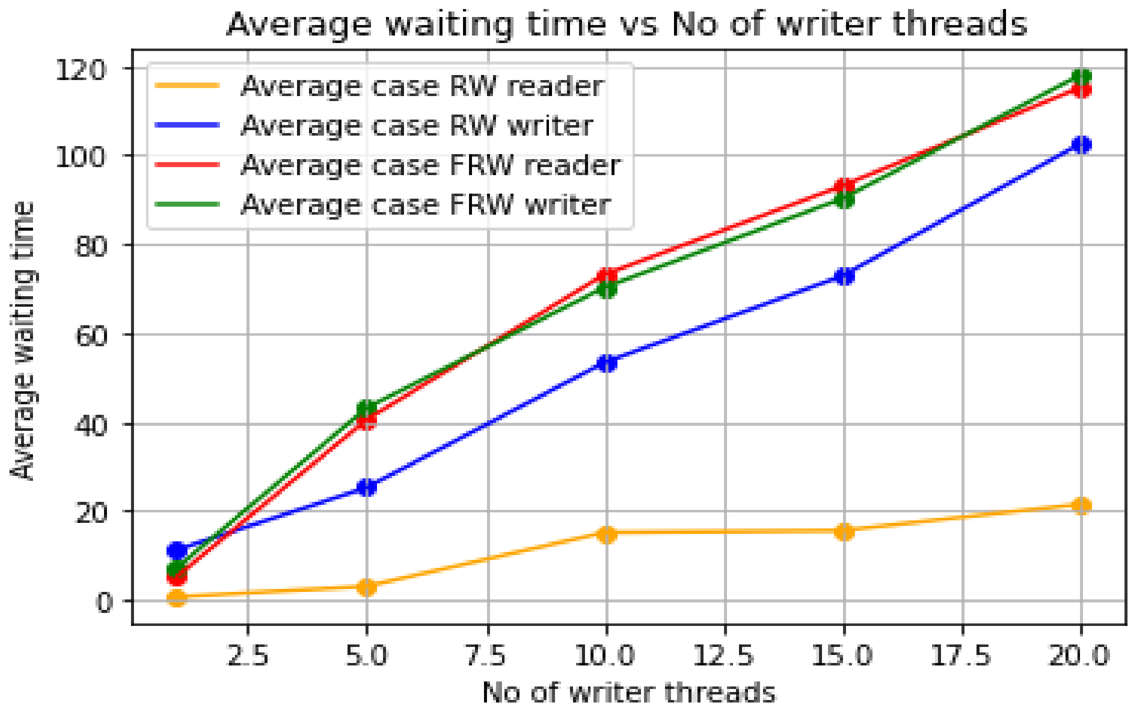
1. Here we use three semaphores rw_mutex,mutex and queue ,each initialized to 1, to attain mutual exclusion and also to solve the bias in normal reader writers algorithm.
2. Rw_mutex and mutex semaphores work similarly as in the normal reader-writers problem and the queue is used to preserve the order in which the requests from threads occurred.

<u>Graphs:</u>

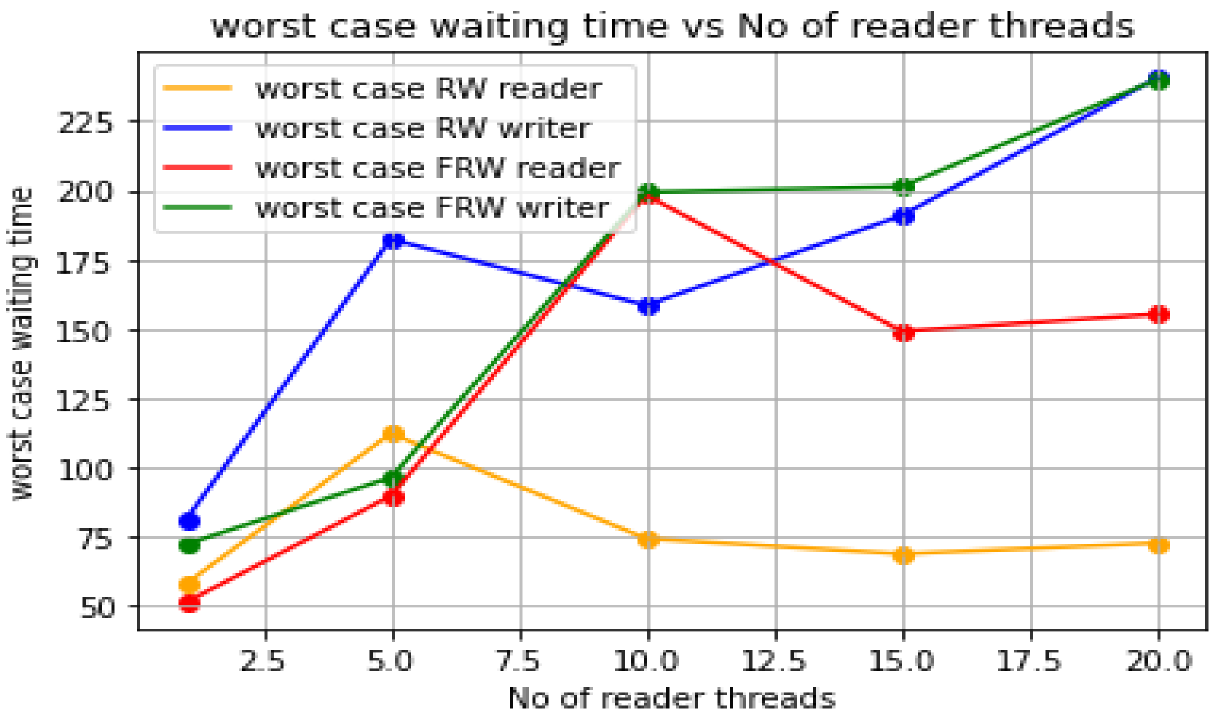
1.


Average waiting time vs No of reader threads
Average waiting time
No of reader threads
Average case RW reader
Average case RW writer
Average case FRW reader
Average case FRW writer
0
20
40
60
80
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0

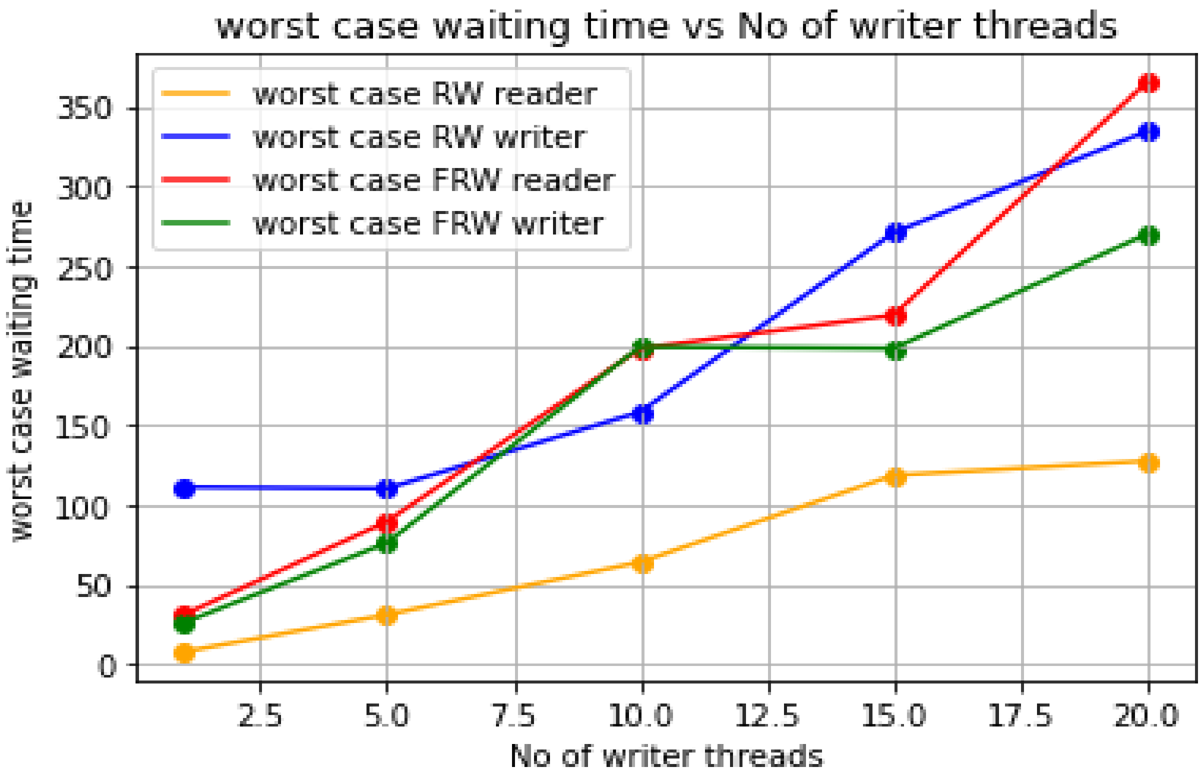
2.


Average waiting time vs No of writer threads
Average waiting time
No of writer threads
Average case RW reader
Average case RW writer
Average case FRW reader
Average case FRW writer
0
20
40
60
80
100
120
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0

3.


worst case waiting time vs No of reader threads
worst case RW reader
worst case RW writer
worst case FRW reader
worst case FRW writer
worst case waiting time
No of reader threads
50
75
100
125
150
175
200
225
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0

4.


worst case waiting time vs No of writer threads
worst case RW reader
worst case RW writer
worst case FRW reader
worst case FRW writer
worst case waiting time
No of writer threads
0
50
100
150
200
250
300
350
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0

Behavior of the graphs:

1. For all the graphs I took average time in critical section and remainder section as 5 and 10 respectively.
2. From the first two graphs we can say that the average waiting time of writer threads is far larger than that of reader threads in biased reader-writers algorithm but in case of fair reader-writers they are almost equal.
3. Also the average waiting times in case of fair reader-writers is somewhat higher than that of normal reader-writers. It may be because of the complexity of the algorithm.
4. From the latter two graphs we can see that the worst case waiting time of writer threads is less in the fair reader writers algorithm than in normal reader-writers algorithm since in normal reader-writers algorithm writer threads may starve due to preference of reader threads.
5. Also in fair RW algorithm worst case waiting times of reader and writer threads are almost equal but in case of biased algorithm worst case waiting times of reader threads are far less than that of writer threads.