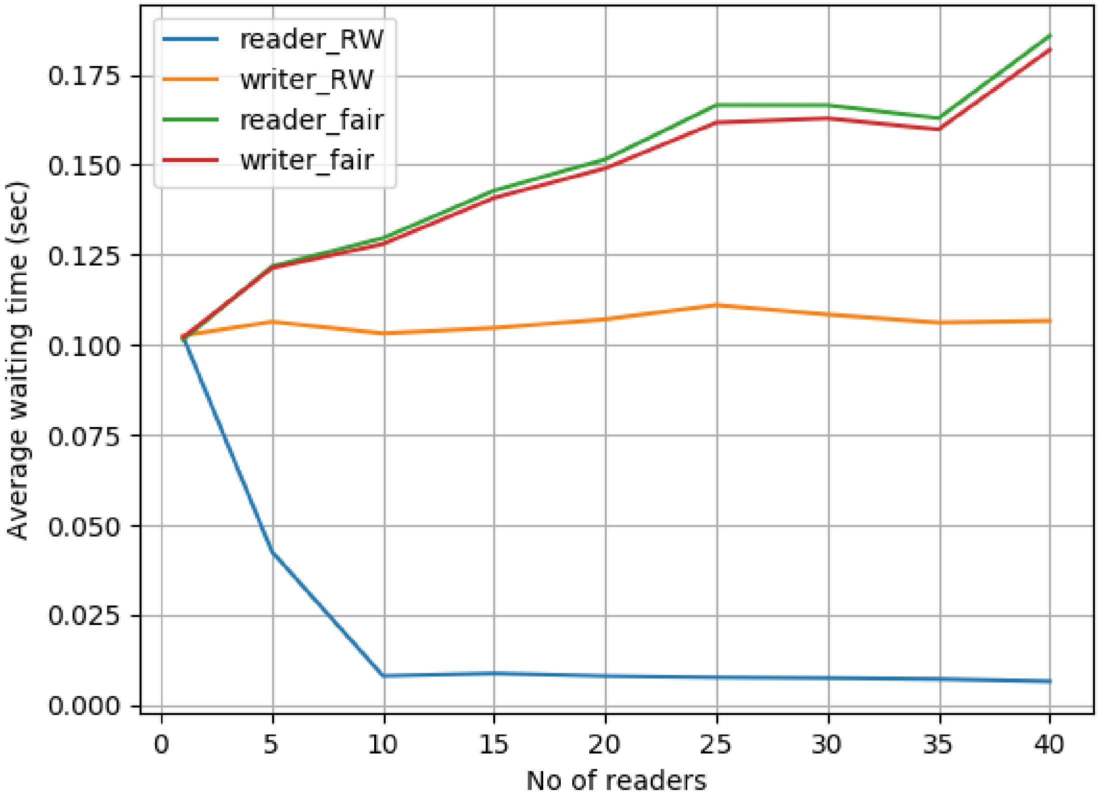
Operating Systems – 2
Report
Assignment – 3
EE17BTECH11041
Sumanth Reddy Cherupally

The output of the program demonstrates the following properties:

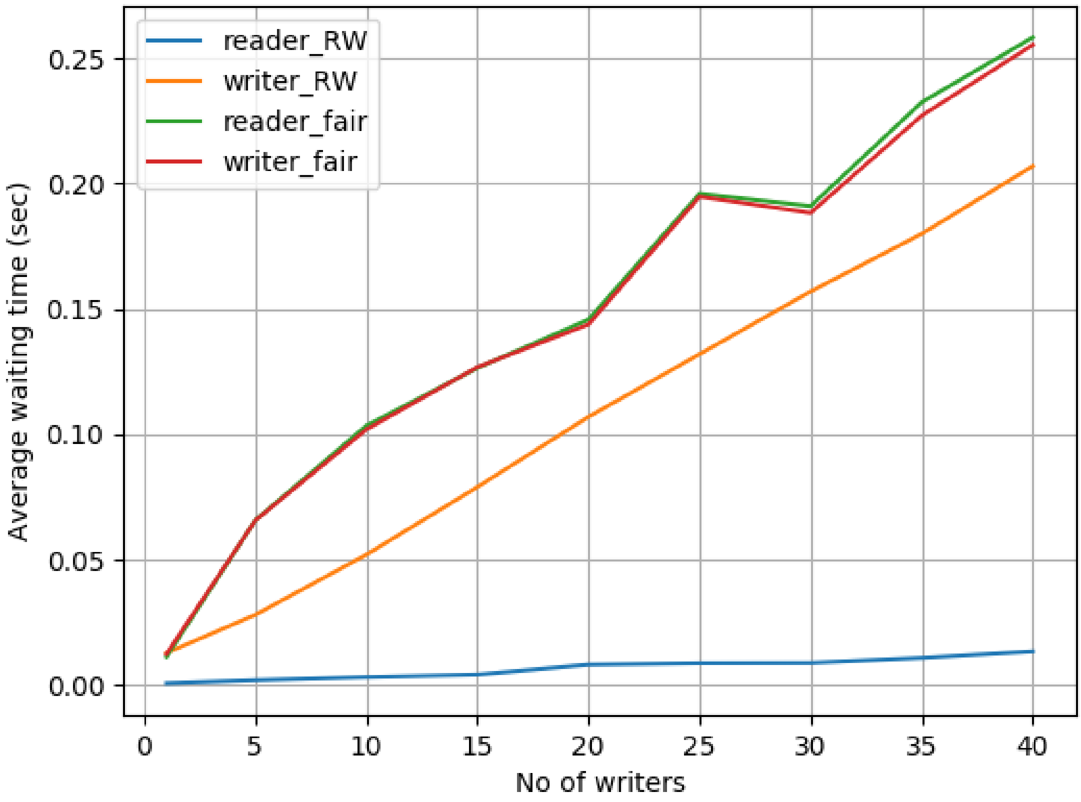
- A reader thread enters the CS only if there is no writer thread in the CS – Mutual exclusion between reader and writer threads.
- But a reader thread can enter the CS even if there are reader threads already in the CS.
- A writer thread enters the CS only if there is no other thread (writer or reader) in the CS.

The required four graphs are below :

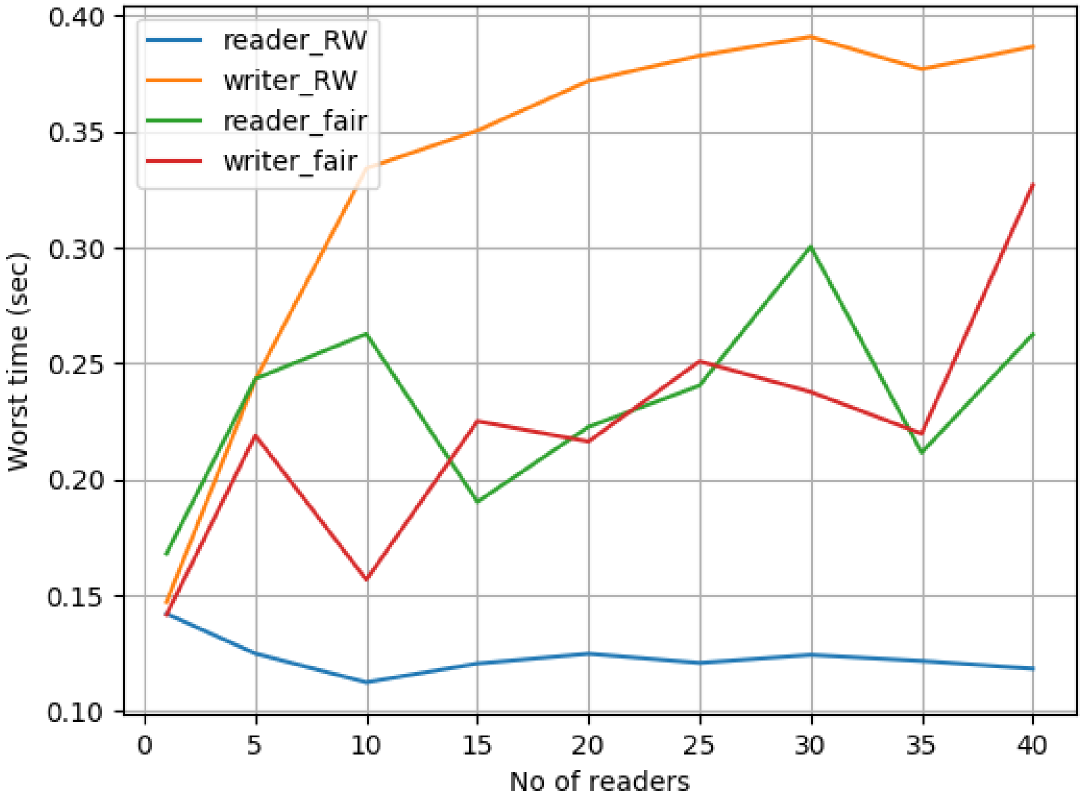
No of readers are varied : Avg waiting time vs No of readers threads (writers = 20)



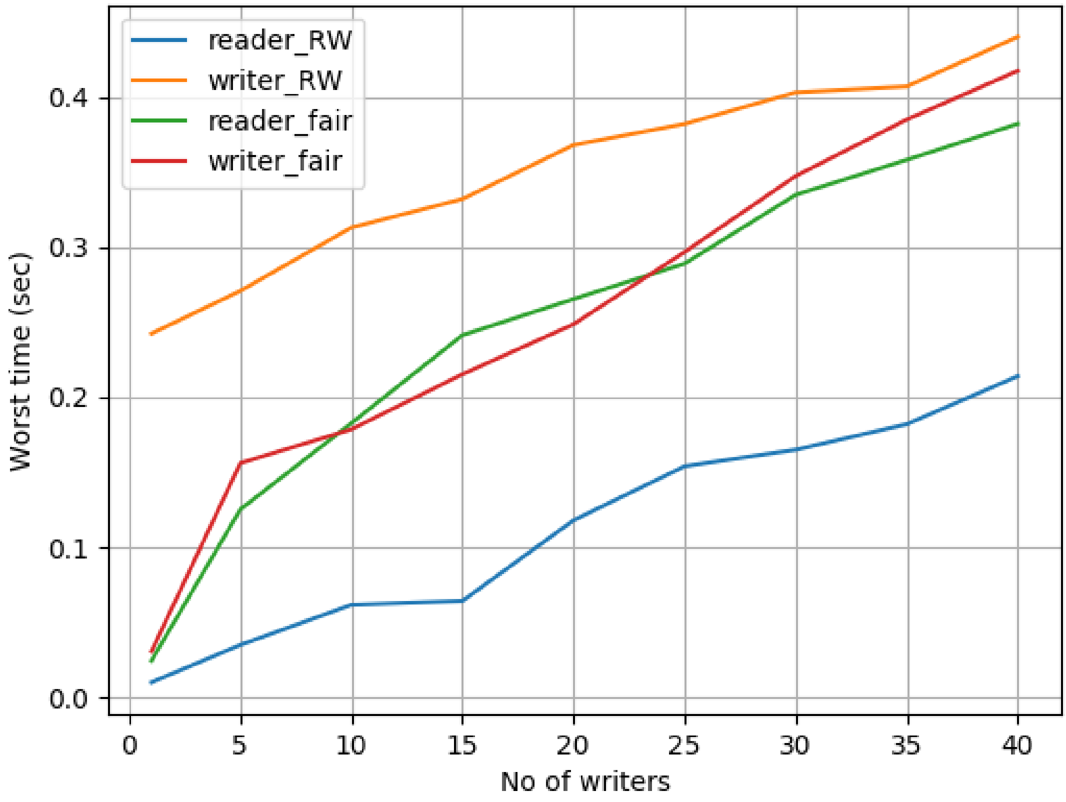
No of writers varied : Avg waiting time vs No of writer threads (readers=20)



No of readers varied : Worst waiting time vs No of readers threads (writers=20)



No of writers varied : Worst waiting time vs No of writer threads (readers=20)

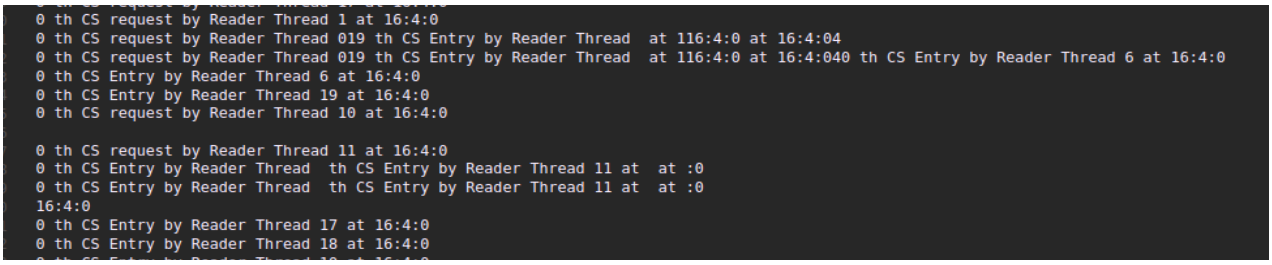


In the first and second figures we can see that the average waiting time for both readers and writers is higher in the fair case than in normal case as the complexity of locking and unlocking the critical section is much higher in the fair case which uses 3 semaphores and 3 other shared variables whereas normal solution uses only 2 semaphores and one shared variable. Thus as no of readers and writers vary the avg waiting time of the normal solution is lower than the fair one.One more point to note is that in the normal case the reader threads have much less waiting time as compared to the writers, whereas in the fair solution the writers and readers have almost the same waiting times. Therefore we can say this solution is fair.

Now coming to the worst waiting times we can clearly see that the normal solution gives more priority to the readers than the writers that is why the worst waiting time of readers is much less than the writers. In fair solution the readers and writers have identical worst waiting times and are less than the writers worst waiting time in the normal solution, which solves the problem of starvation which was happening to the writer threads in the normal solution.

The fair solution gives writers and readers priority to enter the CS based on their arrival(FIFO) whereas it was not the case in the normal solution where the writer threads would starve if the readers kept on entering the CS.

The log file will contain some logs like the one below :



```
0 th CS request by Reader Thread 17 at 16:4:0
0 th CS request by Reader Thread 1 at 16:4:0
0 th CS request by Reader Thread 019 th CS Entry by Reader Thread  at 116:4:0 at 16:4:04
0 th CS request by Reader Thread 019 th CS Entry by Reader Thread  at 116:4:0 at 16:4:040 th CS Entry by Reader Thread 6 at 16:4:0
0 th CS Entry by Reader Thread 6 at 16:4:0
0 th CS Entry by Reader Thread 19 at 16:4:0
0 th CS request by Reader Thread 10 at 16:4:0

0 th CS request by Reader Thread 11 at 16:4:0
0 th CS Entry by Reader Thread  th CS Entry by Reader Thread 11 at  at :0
0 th CS Entry by Reader Thread  th CS Entry by Reader Thread 11 at  at :0
16:4:0
0 th CS Entry by Reader Thread 17 at 16:4:0
0 th CS Entry by Reader Thread 18 at 16:4:0
```

This happens only for reader threads because because multiple reader threads can acess the CS at the same time and they will try to print their log in the file at the same time. Therefore some concurrency issues in writing to the log file.

This will not happen with any writer threads because the are the only ones present in the CS at the time of logging to the file.