

OPERATING SYSTEMS 2 (CS3523)

PROGRAMMING ASSIGNMENT 5 REPORT

NAME: SONGA KOTESH SATVIK

ROLL NO: CS20BTECH11048

The aim of the assignment is to find a solution to the reader writers problem.

Working of the code

- The required values are obtained from the input file
- We create a class thread_stats which contains the stats of each thread.
- A random generator and two exponential distributions of given mean are used to generate values which represent the critical section and the remainder section running times.
- A function updateStats is used which updates the waiting times of the threads.
- Nr (given in input) threads are created and passed to the reader function in which each thread runs for kr times and similarly nw threads are created and passed to the writer function in which each thread runs for kw times.

For the regular reader-writer solution,

Two semaphores rw_mutex and mutex are used which are initialised to 1.

Then,

Writer:

```
For(i=0;i<kw;i++)
{
    Wait(rw_mutex);
    /* writing is performed */
    Signal(rw_mutex);
}
```

Reader:

```
For(i=0;i<kr;i++)
{
    Wait(mutex);
```

```

        Read_count++;
        If(read_count==1)
            Wait(rw_mutex);
        Signal(mutex);
        /* reading is performed */
        Wait(mutex);
        Read_count--;
        If(read_count == 0)
            Signal(rw_mutex);
        Signal(mutex);
    }

```

For the fair reader-writers solution,
Three semaphores rw_mutex, mutex, in are used which are initialised to 1

Writer:

```

For(i=0;i<kw;i++)
{
    Wait(in);
    Wait(rw_mutex);
    Signal(in);
    /* writing is performed */
    Signal(rw_mutex);
}

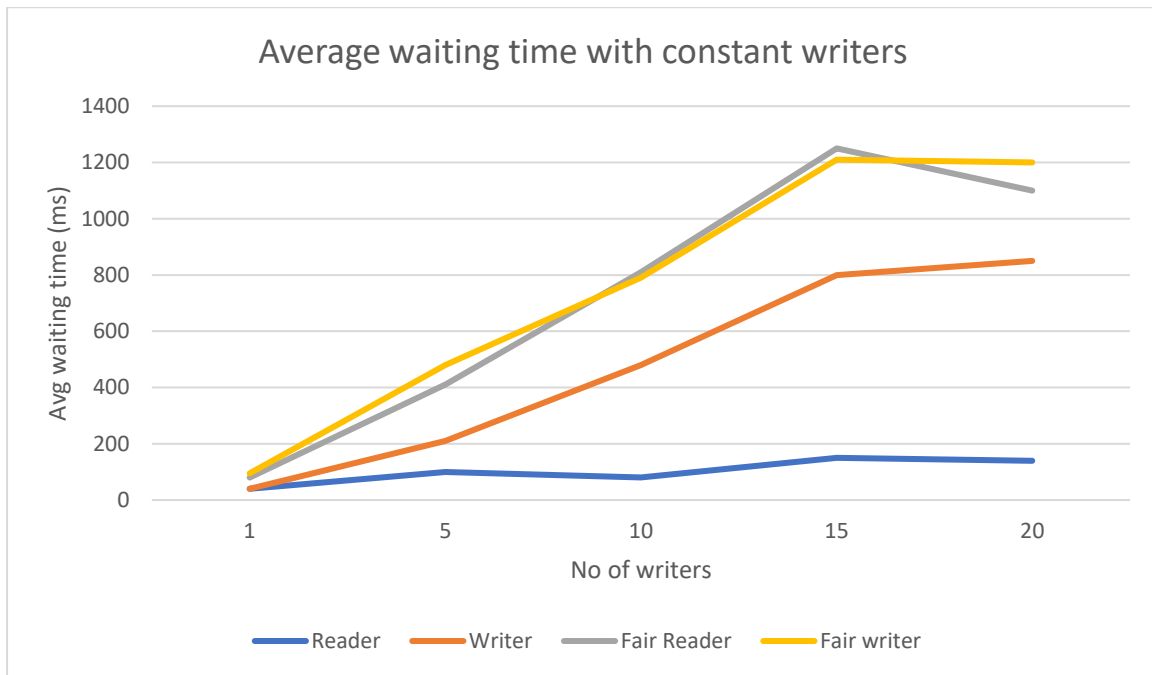
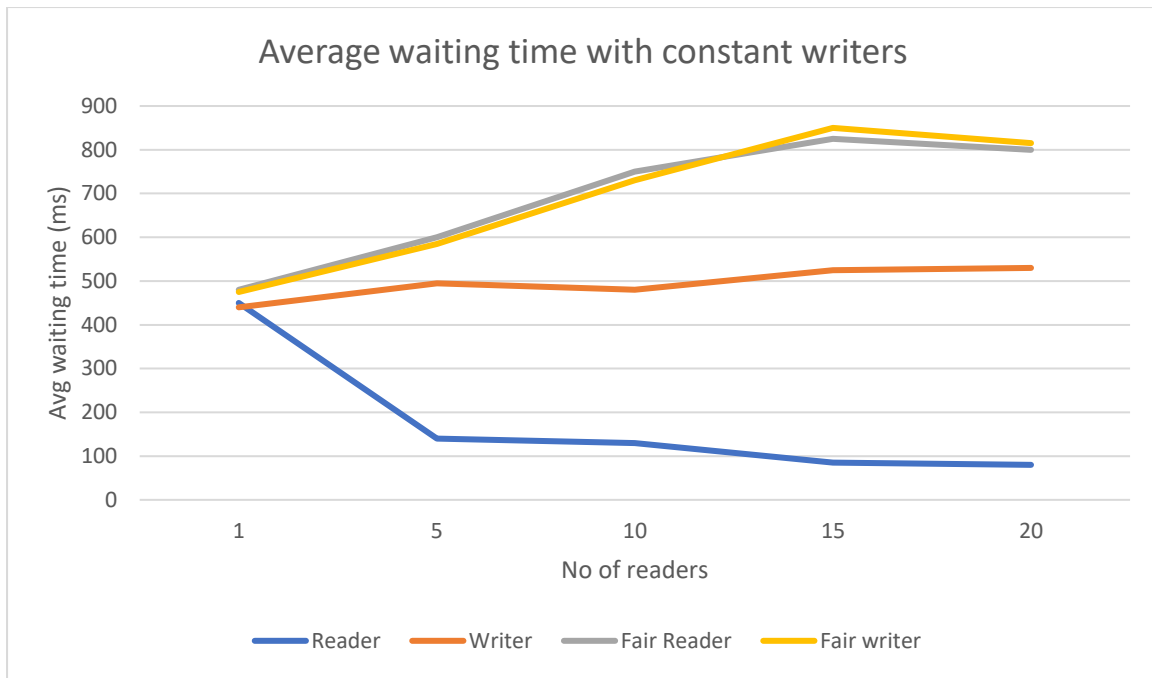
```

Reader:

```

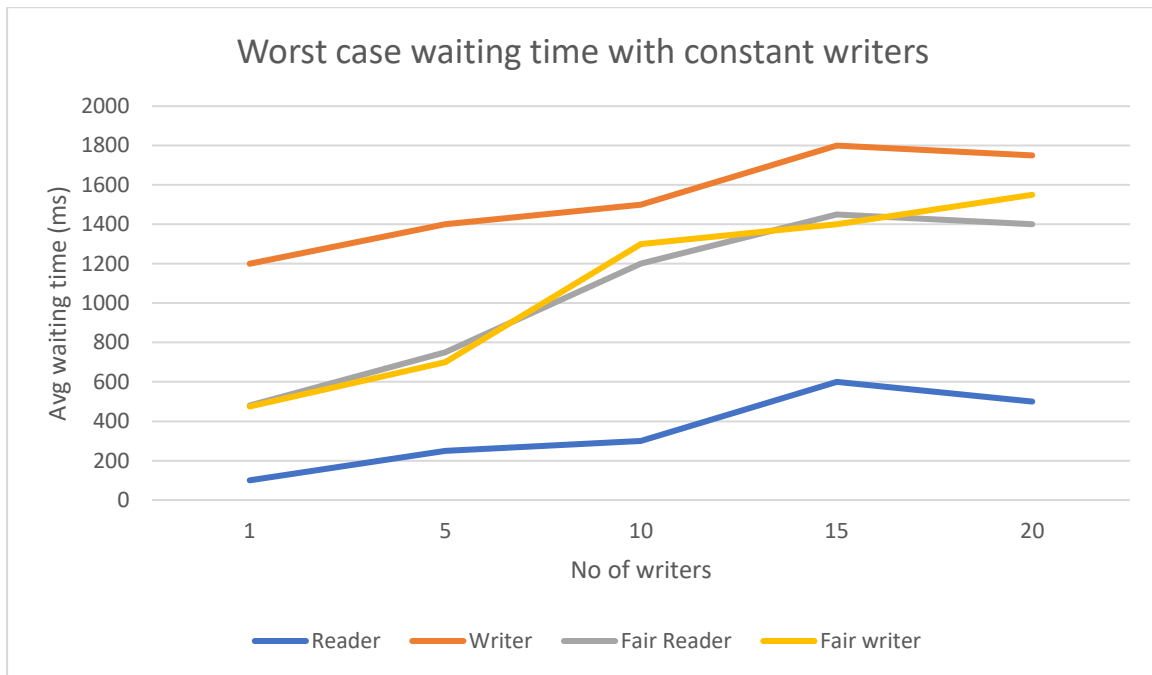
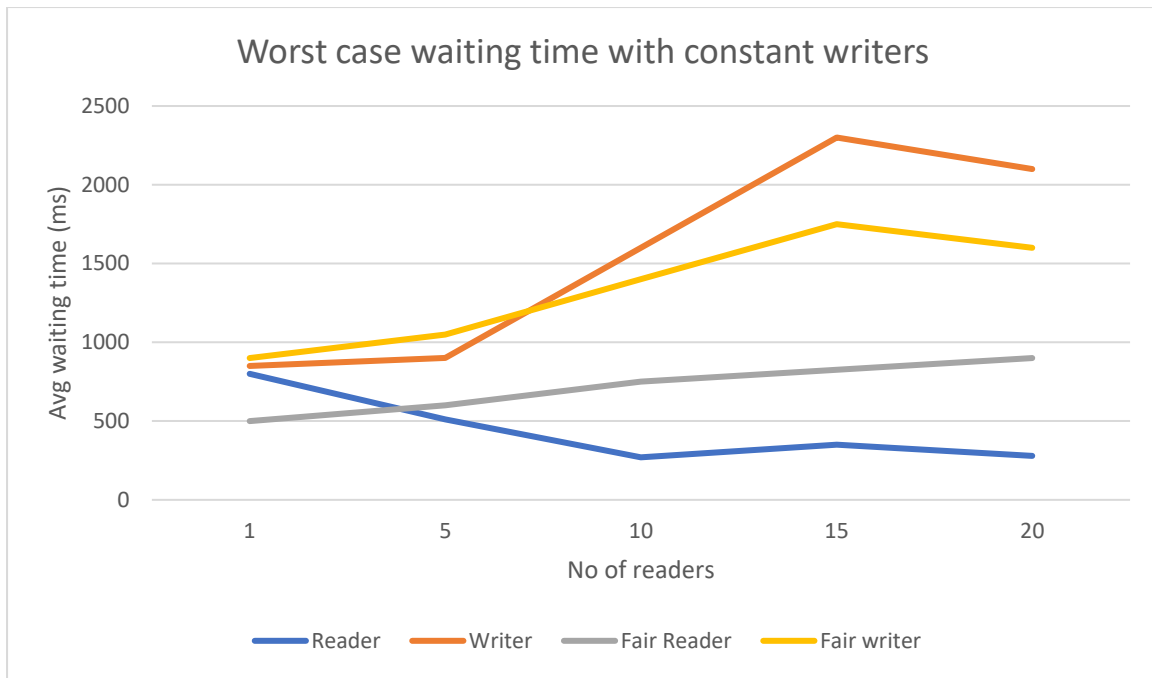
For(i=0;i<kr;i++)
{
    Wait(in);
    Wait(mutex);
    Read_count++;
    If(read_count==1)
        Wait(rw_mutex);
    Signal(in);
    Signal(mutex);
    /* reading is performed */
    Wait(mutex);
    Read_count--;
    If(read_count == 0)
        Signal(rw_mutex);
    Signal(mutex);
}

```



Fair solution ensures that the solution is fair at the cost of increasing the average waiting time of the threads.

The average time of the readers thread is the least.



Fairness also increases the longest waiting times of the threads.

Therefore we can see that fairness increases the waiting times but reduces the possibility of starvation.