OS-2 Programming Assignment-5 Report

CS20BTECH11009

Input:

This programs takes the input as “inp-params.txt” in which it has nw(no of writer threads), nr(no of reader threads) ,kw (no of times of each writer thread) , kr (no of times of each writer thread) , mcs , mrt.

Code Design :

1) Irrescpective of type of the code here most of the code remains the same. Only the lock and release section will be different.first we will see that. Necessary inputs are taken . nw,nr,kw,kr ,mcs , mrt

2) Then we will call a function which will create the writer and reader threads and we give the reader function as parameter for reader threads,we give the writer function as parameter for writer threads, . And in the function itself the threads are joined and the threads were deleted. And in the same function the semaphores are initialised to 1 and at the end they were destroyed.

3) We declare 2 double pointers used as 2 d arrays which are dynamically allocated and is used to store waiting times for each writer and reader thread.

4)Now we come to the function reader and writer . In that index is passed as argument. The t1, t2 are passed as arguments for sleep function.In this function a for loop is there which runs kw , kr times respectively for writer and reader function .

5) In each loop, at the start the request time (time at which process request) is noted and then entry section is called and now after the entry section the entry time (the time at which it enters the critical section) is noted . Sleep (t1/1000) critical section. Sleep (t2 /1000) remainder section.

6) Now we will see the lock section and release section in reader and writer function in both readers preference and fair reader writer problem.

7)

Readers preference : semaphores wr_mutex , mutex1 intilalised to 1

a) Reader function :

- Lock section : mutex1 is locked and when reading_count is available it is increased by 1 . If it is the first reader it will wait on wr_mutex. If its not the first reader the signal on mutex1 is called.
- Release section : mutex1 is locked and when reading_count is available it is decreased by 1. and if no reader is there , signal on wr_mutex is called . If some more readers are left then signal on mutex1 is called.

b) Writer function:

- Lock section : locked using sem_wait(&wr_mutex)
- Release section : released using sem_post(&wr_mutex)

Fair Reader-writer : semaphores wr_mutex ,mutex1,queueorder to 1

a) Reader function :

- Lock section : almost same as in readers preference . Slight change is before the mutex1 is locked the sem_wait(&queueorder) is called.and after checking the no of readers the signal on queueorder is called and then signal on mutex1.
- Release section : same as in readers preference

b) Writer function:

- Lock section :intially queueorder is locked and then wr_mutex is locked and then queueorder is released.
- Release section : same as in readers preference

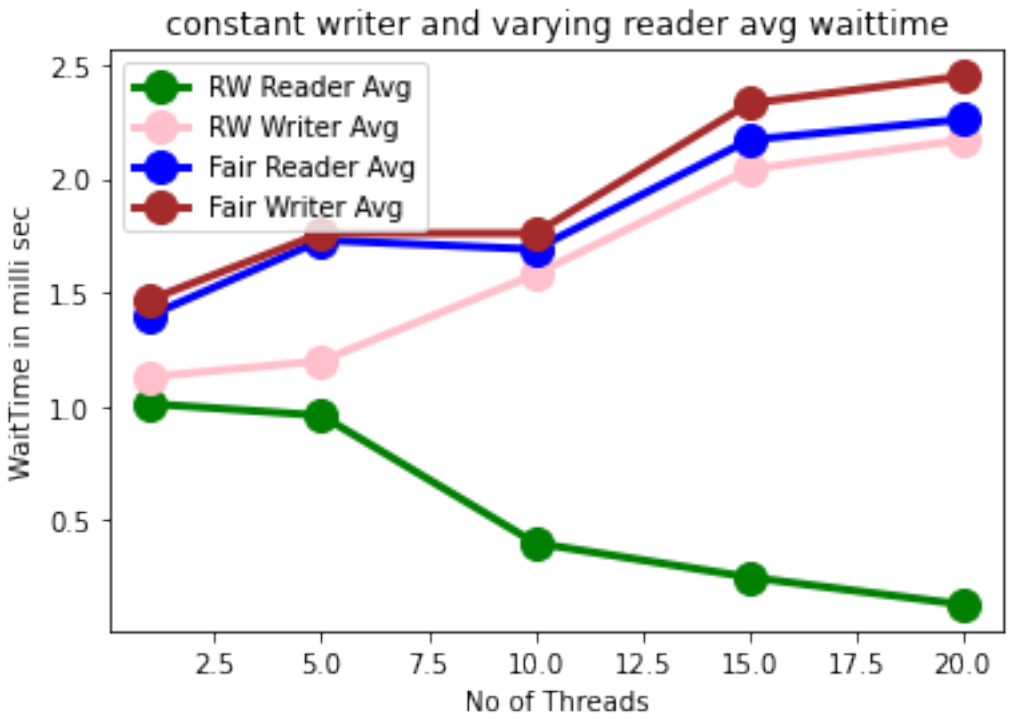
Output :

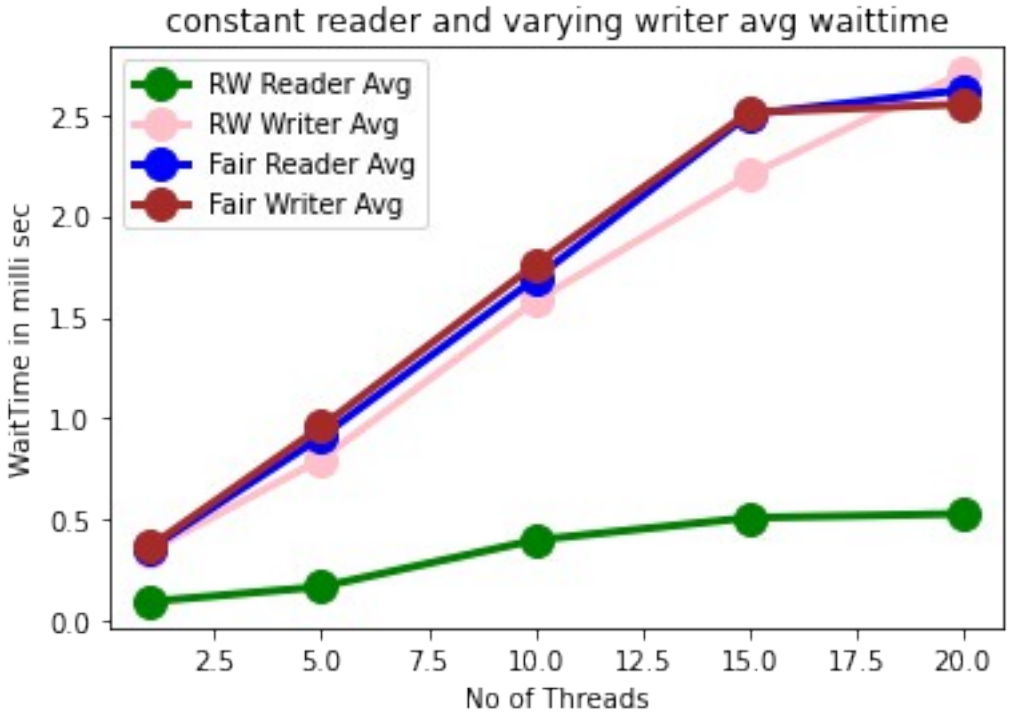
1) For each code it will generate a Log.txt in which it has information about at what time it has entered and and at what request has made , and at what time it has exited the system.

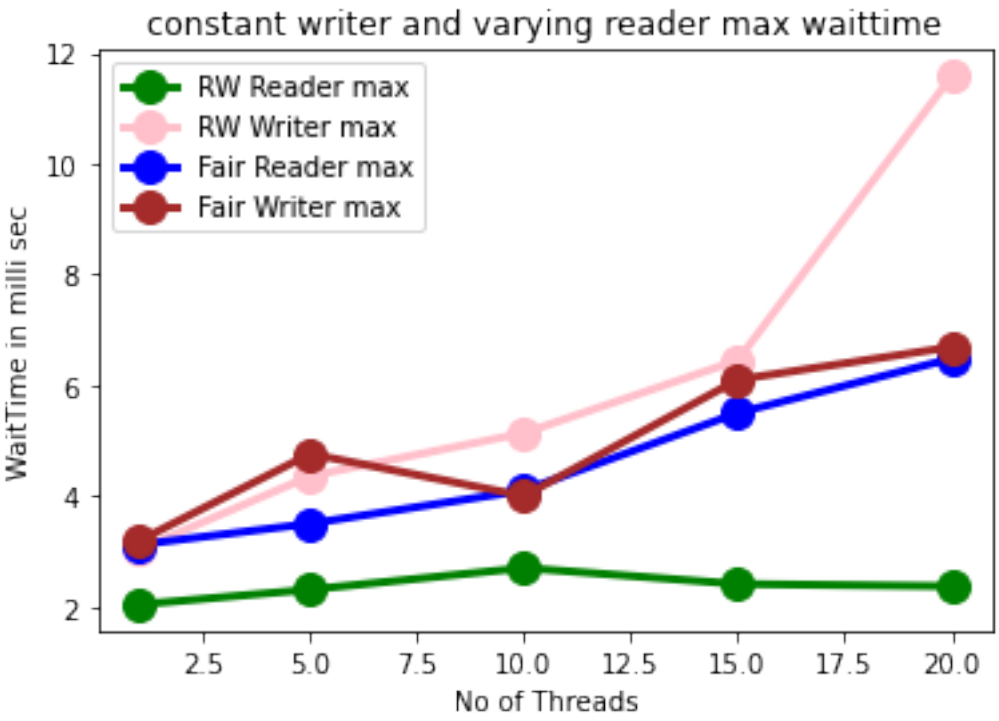
2) For each code it will generate a Avg_time.txt in which it has information about average time for each writer and reader thread , and also average waiting time , max waiting time for all writer
and reader threads

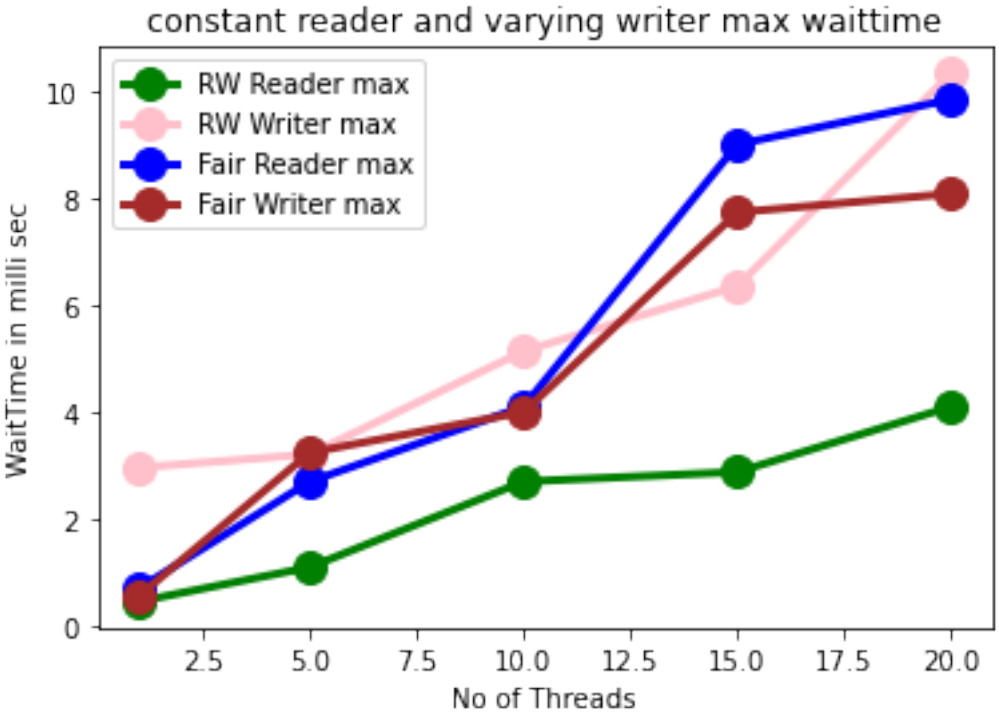
Graphs :

kr = 10 , kw = 10 , mcs =5 , mrt = 20. No of threads from 1 to 20 with increase of 5.






constant writer and varying reader max waittime
RW Reader max
RW Writer max
Fair Reader max
Fair Writer max
WaitTime in milli sec
No of Threads
12
10
8
6
4
2
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0


constant reader and varying writer max waittime
RW Reader max
RW Writer max
Fair Reader max
Fair Writer max
WaitTime in milli sec
No of Threads
10
8
6
4
2
0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0

Analysis :

- In graph 2 , the fair reader and fair writer have almost same values. As preference for reader and writer is almost same.
- The fair reader writer will make sure the starvation does not occur but the waiting times for both readers and writers have increased
- In graph 1 for normal reader the avg waiting time goes on decreasing as the no of threads is increasing as the no of threads increases the reader prefernce is given waiting time decreases
- In graph 1 , normal writers have much higher waiting time than the normal readers
- In graph 1 the fair values are higher than normal values
- In graph2 the fair values and the normal writer have near values than compared to normal writer values
- expect in graph1 for normal reader ,all the values are increasing with increase in no of threads.
- In graph3 the normal writer have more worst waittime than the fair reader and writer.