Assignment 3 Report Operating Systems

Need to Implement Readers-Writers problem and Fair Readers-Writers problem using Semaphores in C++.

1st Program involves setting up a semaphore with a counter initialised to a number of readers allowed simultaneously read the critical section. A writer then sequentially reduces the semaphore counter by that number by waiting until all readers exit CS and at the same time not allowing new readers to enter CS. This is achieved using locks. This can lead to starvation as a Writer thread does not have a chance to execute while any number of Readers continuously entering and leaving the Critical Section.

For 2nd program Starvation can be avoided by using one more semaphore which both reader and writer need to acquire before entering the Critical Section. The semaphore queue is Fair and gives turns based on order they are waiting.This is clearly indicated in the graph as waiting time for writer increases.

The main complication was to correctly implement semaphores else the process goes into a infinite loop.

The Graph comparing the Average Waiting time is shown below.