

## Complie and Execute

Q1: `gcc A2Q1.c -o a2q1 -lpthread` and `./a2q1 30 60`

Q3: `gcc A2Q3.c -o a2q3 -lpthread` and `./a2q3 30 60`

## Question 1

## Measurement

```
yma67@teaching:~/ws/ece427/a2$ gcc A2Q1.c -o a2q1 -lpthread
yma67@teaching:~/ws/ece427/a2$ ./a2q1 30 60
===== [Reader Preference] =====
[Reader]>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
[Max wait] 12.000000 ms
[Min wait] 0.000000 ms
[Avg wait] 0.010708 ms
[Count]    30000
[Writer]>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
[Max wait] 3291.000000 ms
[Min wait] 0.000000 ms
[Avg wait] 109.443333 ms
[Count]    300
```

## Observation

This implementation have a starvation problem that writer have to wait significantly longer than reader.

## Reasoning

- All readers at a time slice should be considered to hold the `rw_mutex` lock if one of them actually holds, and therefore blocks all other pending writers
- Writers have to wait for readers if there are multiple readers, even if some readers arrive after the writer
- For instance, if one reader is reading, and while the reader is reading, a writer followed by another  $n$  readers came, then the writers will wait for the (at least)  $n+1$  readers to finish their reading (even if  $n$  readers came after the writer), since the writer will not give up file mutex lock until `read_count` is 0.

## Question 2

Starvation already observed in Q1

### Question 3

See A203.c

## Question 4

## Measurement

```
yma67@teaching:~/ws/ece427/a2$ gcc A2Q3.c -o a2q3 -lpthread
yma67@teaching:~/ws/ece427/a2$ ./a2q3 30 60
===== [Fair] =====
[Reader]>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
[Max wait] 265.000000 ms
[Min wait] 0.000000 ms
[Avg wait] 4.448300 ms
[Count]    30000
[Writer]>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
[Max wait] 242.000000 ms
[Min wait] 0.000000 ms
[Avg wait] 5.916667 ms
[Count]    300
```

Since 4.45 ms is not significantly higher than 5.91 ms, there is no starvation occurred.

### Proof

$$\text{std\_dev\_read} = \text{range} / 4 = 265 / 4 = 66.25 \text{ (approx)}$$
$$\text{std\_dev\_write} = \text{range} / 4 = 242 / 4 = 60.5 \quad (\text{approx})$$

using Unpaired T-test, p value is 0.7022, and even if alpha is 0.1, it is still considered as **not significant**.

**END of this assignment**