

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING



Yashpal Singh Shekhawat

11801730

B36

K18KK

School of Computer Science and Engineering
Lovely Professional University Phagwara, Punjab

Name: Yashpal Singh Shekhawat

Student Id : 11801730

Email: shekhawatyash321@gmail.com

Github link: https://github.com/11801730/reader_and_writer/blob/master/main.c

Problem Statement: Q1. Write a multi-threaded C program that gives readers priority over writers concerning a shared (global) variable. Essentially, if any readers are waiting, then they have priority over writer threads -- writers can only write when there are no readers.

Constraints:

Multiple readers/writers must be supported (5 of each is fine)

Readers must read the shared variable X number of times

Writers must write the shared variable X number of times

- Readers must print:
 - The value read
 - The number of readers present when value is read
- Writers must print:
 - The written value
 - The number of readers present were when value is written (should be 0)
 - Before a reader/writer attempts to access the shared variable, it should wait some random amount of time

Note: This will help ensure that reads and writes do not occur all at once

Concept used

- pthreads
- mutexes
- condition variables to synchronize access to the shared variable

Output:

```
Enter the Initial value:
1
_____
no. of Reader:
2
R0
R1
_____
no. of Writer:
1
W0
_____
_____
W0 Wait for Random time= 1
Enter the no. of time W0 want to write:
1
Now W0 is writing...
Enter the 1th value to write:
2
UPDATED value of Shared variable to 3
_____
R0 wait for Random time = 1
R1 wait for Random time = 1
Enter the no. of time R0 want to read:
Enter the no. of time R1 want to read:
1
Now R0 is reading...
R0 read the shared value = 3
No. of Readers present = 2
1
Now R1 is reading...
R1 read the shared value = 3
No. of Readers present = 1
-----After joining the threads-----
Final value of share variable = 3
-----
Process exited after 36.41 seconds with return value 34
Press any key to continue . . .
```