# Operating Systems 2
Name :- Ajinkya Bokade
Roll No :- CS17BTECH11001

## Aim
To implement the bounded buffer producer−consumer problem using Semaphores and Locks.

## Design of Program
The input parameters to the program will be a file, named inp−params.txt, consisting of all the parameters described above:capacity (size of buffer), np (number of producer threads), nc (number of consumer threads), cntp (count of producer threads), cntc (count of consumer threads), μp (Average value for exponential distribution of delay value t1 in producers), μc (Average value for exponential distribution of delay value t2 in consumers).

The np number of producer and nc number of consumer threads are created using pthreads.
In sempahore implementation, 2 semaphores full, empty are initialized to value of 0 and buffer capacity and a binary semaphore mutex is used.
Mutex semaphore provides mutual exclusion.
The semaphore implementation is same as that of Bounded Buffer problem using semaphores .
Average waiting time for producers are calculated by adding the differences between the time when thread puts the produced item in buffer and time when it made a request to put item in buffer and dividing by count of producers (cntp).
Similarly, Average waiting time for consumers are calculated by adding the differences between the time when thread consumes the item from buffer and time when it made a request to consume item in buffer and dividing by count of producers (cntc).
In lock implementation, before accessing buffer thread acquires the lock and after producing item and putting in buffer in case of producers and after consuming item from buffer in case of

consumers, lock is released. Producers dont produce if buffer is full and consumers dont consume if buffer is empty.
Average time for producers and consumers are calculated in similar way as that in semaphore implementation.

## Comparison of the performance of producer and consumer threads

Input parameters :− Buffer Capacity = 100
Number of producer threads (np) = 10
Number of consumer threads (nc) = 15
Count of producer threads (cntp) = 15
Count of consumer threads (cntc) = 10
These parameters are kept constant throughout.
Only μp (Average value for exponential distribution of delay value t1 in producers), μc (Average value for exponential distribution of delay value t2 in consumers) are varied.
The ratio of μp/μc consists of the following 11 values: 10, 8, ..... 2, 1, 0.8, 0.6, ..... 0.2, 0.1.

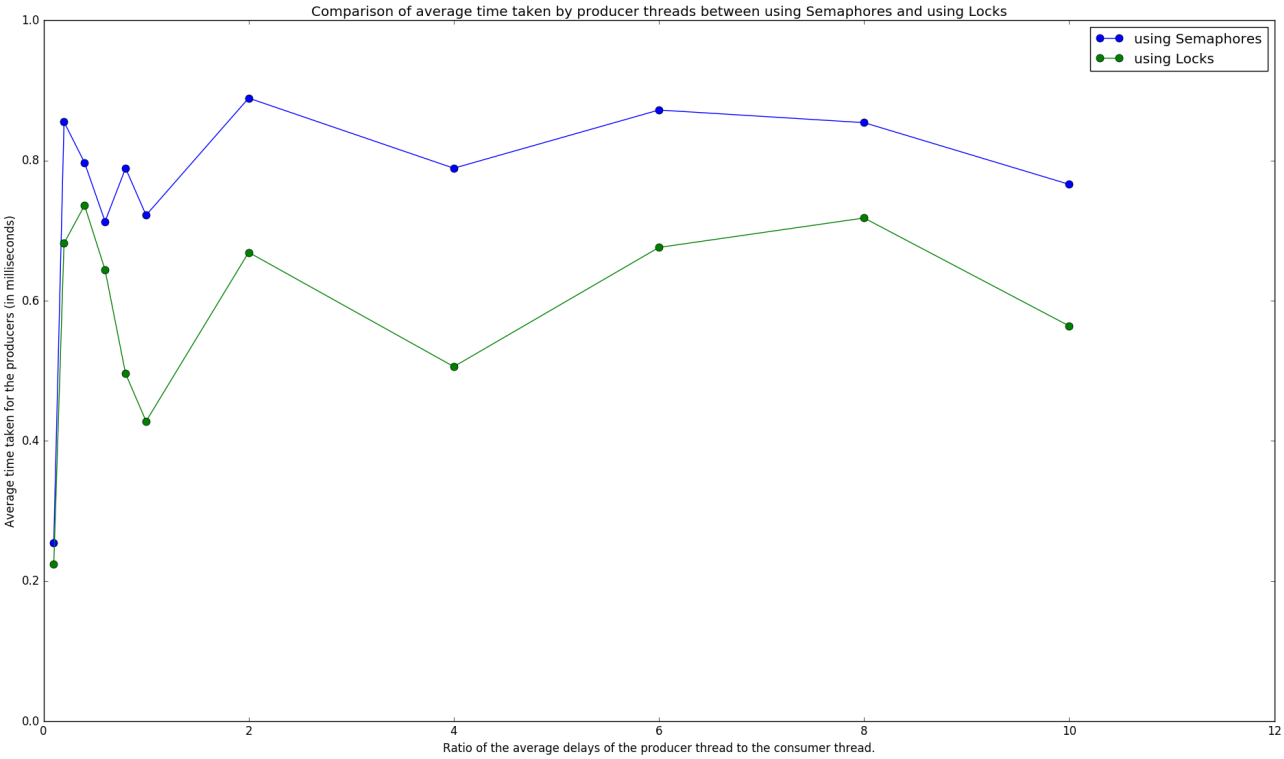
Average waiting time for Producers and Consumers

| μp/μc | Average Time taken by Producers in semaphores (in milliseconds) | Average Time taken by Producers in locks (in milliseconds) | Average Time taken by Consumers in semaphores (in seconds) | Average Time taken by Consumers in locks (in seconds) |
|---|---|---|---|---|
| 10 | 0.766 | 0.564 | 234.378 | 195.829 |
| 8 | 0.854 | 0.718 | 184.333 | 150.385 |

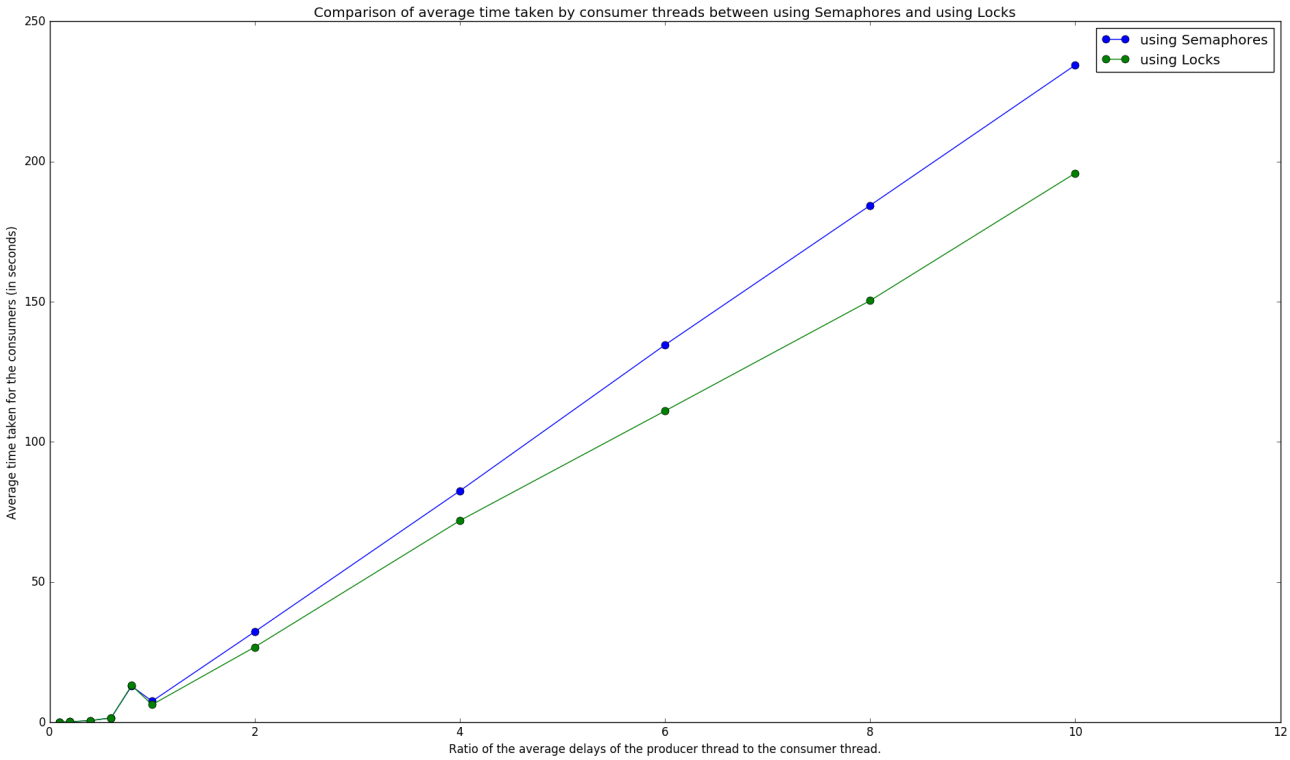
| | | | | |
|---|---|---|---|---|
| 6 | 0.872 | 0.676 | 134.584 | 111.045 |
| 4 | 0.789 | 0.506 | 82.529 | 71.93 |
| 2 | 0.889 | 0.669 | 32.304 | 26.836 |
| 1 | 0.722 | 0.428 | 7.537 | 6.316 |
| 0.8 | 0.789 | 0.496 | 12.938 | 13.149 |
| 0.6 | 0.713 | 0.644 | 1.583506 | 1.583103 |
| 0.4 | 0.797 | 0.736 | 0.647594 | 0.647170 |
| 0.2 | 0.855 | 0.682 | 0.212736 | 0.212791 |
| 0.1 | 254.893 | 224.043 | 0.149131 | 0.149038 |

# Graphs

Average time taken for the producers threads vs Ratio of average delays of producer thread to consumer thread



Average time taken for the consumer threads vs Ratio of average delays of producer thread to consumer thread




# Conclusion

We observe that the average time taken by producer threads in lock implementation is less as compared to semaphore implementation and each of them is less than corresponding average time taken by consumer threads. Also, we observe that the average time taken by consumer threads in lock implementation is less as compared to semaphore implementation and it increase with increase of ratio of average delays of producer thread to consumer thread.