CSD204 - OS - Lab06

Lalit Maurya, 2310110164

Analysis of different Mutual Exclusion Algorithms in C++

For this assignment, I wrote a script `q01.cpp` that reads from an input file and produces three output files for the following algorithms:

- TAS
- CAS
- Bounded CAS

The program accepts

- `n`: Number of threads
- `k`: Number of iterations for critical section
- λ_1: Exponential Distribution Argument 1
- λ_2: Exponential Distribution Argument 2

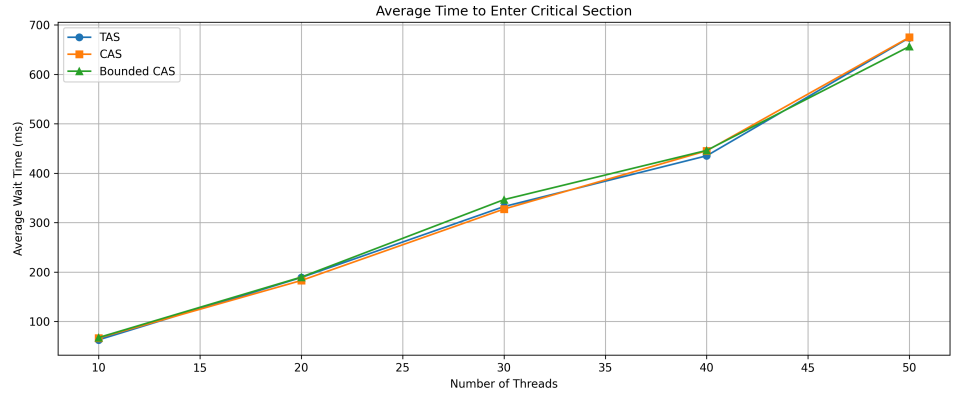
It then creates multiple locks and a log file for each algorithm. By varying the `input.txt` file and rerunning, we can benchmark the different algorithms.

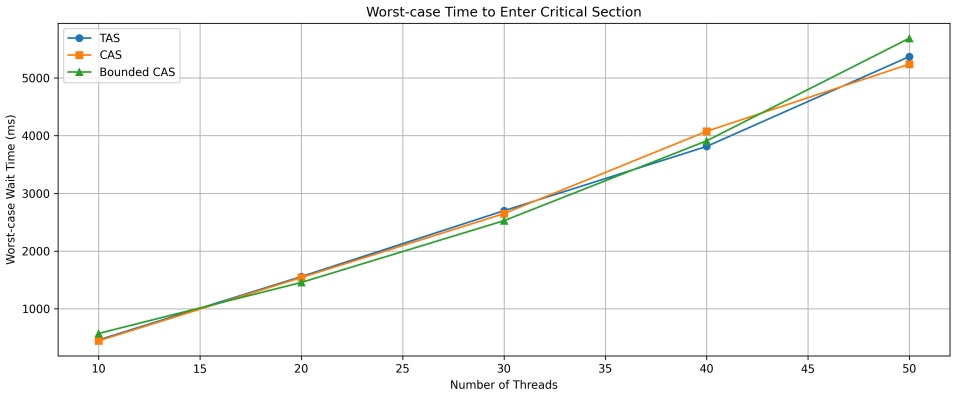
Since the thread count had to be varied from 10 to 50 and 5 reruns were necessary at each go, I opted to create a python script to handle rerunning the script for me.

Outputs

Throughout the tests:

- `k` = 10
- λ_1 = 10
- λ_2 = 10





Waiting Times

Threads	TAS Avg	TAS Worst	CAS Avg	CAS Worst	Bounded CAS Avg	Bounded CAS Worst
10	62.75	460.4	66.456	445.4	67.4	572.2
20	188.879	1557.4	182.85	1540.8	189.719	1456.2
30	332.436	2698.8	327.387	2648.8	346.456	2526.0
40	435.189	3812.8	445.2925	4074.8	446.088	3909.8
50	674.0824	5369.0	674.9644	5238.0	656.3624	5687.2

Performance Comparision

TAS performs best for low thread counts in terms of average waiting time CAS shows most fair behaviour (smallest worst-case wait time)

Conclusion

- Where raw performance is critical and thread count is low, TAS is preferable
- Where fairness is critical, Bounded CAS is preferable