Programming Assignment 3 : Implementing TAS, CAS and Bounded Waiting CAS Mutual Exclusion Algorithms

CO21BTECH11004

In output file time is printed in microseconds which is the difference between Time measured and start time.
Start time is calculated at the beginning of the program.

Global Variables : -

- n,k :- to store the number of threads and number of critical sections to create.
- lamda1, lamda2 :- for getting sleep time (t1 and t2) using exponential distribution
- AvgTime and WorstTime arrays which are used in calculating the average and worst entering time by each thread
- Pointer to output file
- Random generator seed
- Atomic lock

Function declared :- testCS(int threadID)

- Given in assignment
- Found t1 and t2 using exponential distribution, lamda1 and lamda2.
- Entered code for TAS, CAS using inbuilt function and Bounded CAS using inbuilt function and method given in book.

main():-

- n, k, lamda1, lamda2 are read from the input file.
- Dynamically allocate memory to global array.
- Open file pointer to output file
- n threads are created that run testCS and pass thread number as input
- Created threads are joined

Implementation of TAS: -
Used inbuilt function for TAS

```
while (atomic_flag_test_and_set(&lock_stream)) {
    // do nothing
}
```

Implementation of CAS: -
Used inbuilt function for CAS

```
while (true){
    int expected = 0;
    if (atomic_compare_exchange_strong(&lock_stream, &expected, 1)) break;
}
```

Implementation of Bounded-CAS :-
Used inbuilt compare and swap, with method given in book to satisfy bounded waiting.

```
// Bounded CAS implemented from os-book
waiting[thread_id-1] = true;
key = 1;
while (waiting[thread_id-1]==true
 && key == 1){
    int expected = 0;
    if (atomic_compare_exchange_strong(&lock_stream, &expected, 1)) key=0;
}
waiting[thread_id-1] = false;
```

```
int j = (thread_id)%n;
while ((j != thread_id-1) && (!waiting[j])){
    j = (j+1)%n;
}
if (j == thread_id-1) lock_stream = 0;
else waiting[j] = false;
```

Comparison between different algorithms: -
Average Waiting Time varying with number of threads

TAS

No. of threads	Time_taken in each iterations (TAS) (microseconds)					Average Waiting
	1st	2nd	3rd	4th	5th	
10	389.06	365.59	396.39	328.22	360.97	368.046
20	83015.5	96024.5	72955.3	71930	79730.3	80731.12
30	153928	171526	174046	131255	184808	163112.6
40	252350	277159	228383	237501	268779	252834.4
50	363395	340621	362801	365113	301285	346643

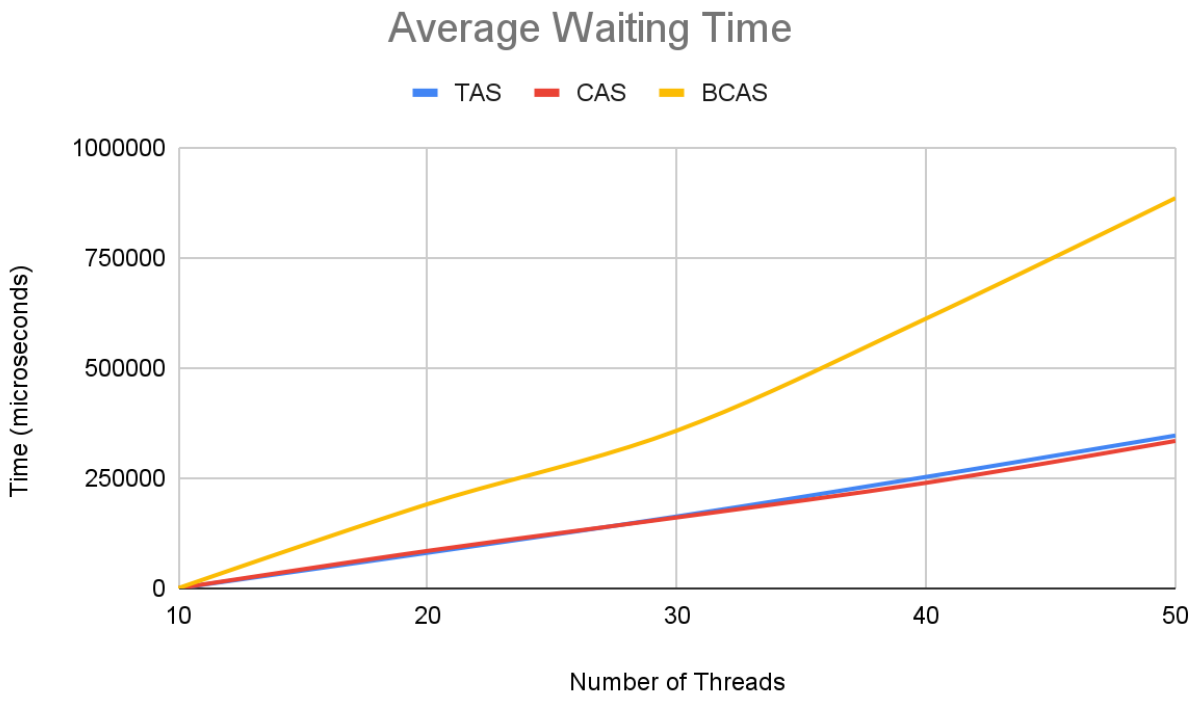
CAS

No. of threads	Time_taken in each iterations (CAS) (microseconds)					Average Waiting
	1st	2nd	3rd	4th	5th	
10	328.44	363.72	380.93	329.69	371.46	354.848
20	81040.3	81499.6	98670.4	75697.6	88358.1	85053.2
30	165598	168348	169203	152539	148282	160794
40	272738	245404	225593	229140	224242	239423.4
50	343917	317002	338354	342685	330223	334436.2

Bounded CAS

No. of threads	Time_taken in each iterations (Bounded CAS) (microseconds)					Average Waiting
	1st	2nd	3rd	4th	5th	
10	492.62	457.21	469.19	481.3	497.64	479.592
20	188184	199976	183566	173570	210218	191102.8
30	386978	324043	392124	382183	303830	357831.6
40	578247	657450	589901	615315	619084	611999.4
50	899075	900918	895396	866721	863528	885127.6

Graph :- (TAS vs CAS vs Bounded CAS)



Worst Waiting Time varying with numbers of threads : -
TAS

No. of threads	Time_taken in each iterations (TAS) (microseconds)					Worst Waiting
	1st	2nd	3rd	4th	5th	
10	4051	3865	3732	3431	3779	3771.6
20	695953	747996	553533	676867	821850	699239.8
30	1869684	1527059	1572158	1681551	1108050	1551700.4
40	2916140	2747889	2304004	2071816	3008113	2609592.4
50	3292006	4046882	2619999	4043994	3608980	3522372.2

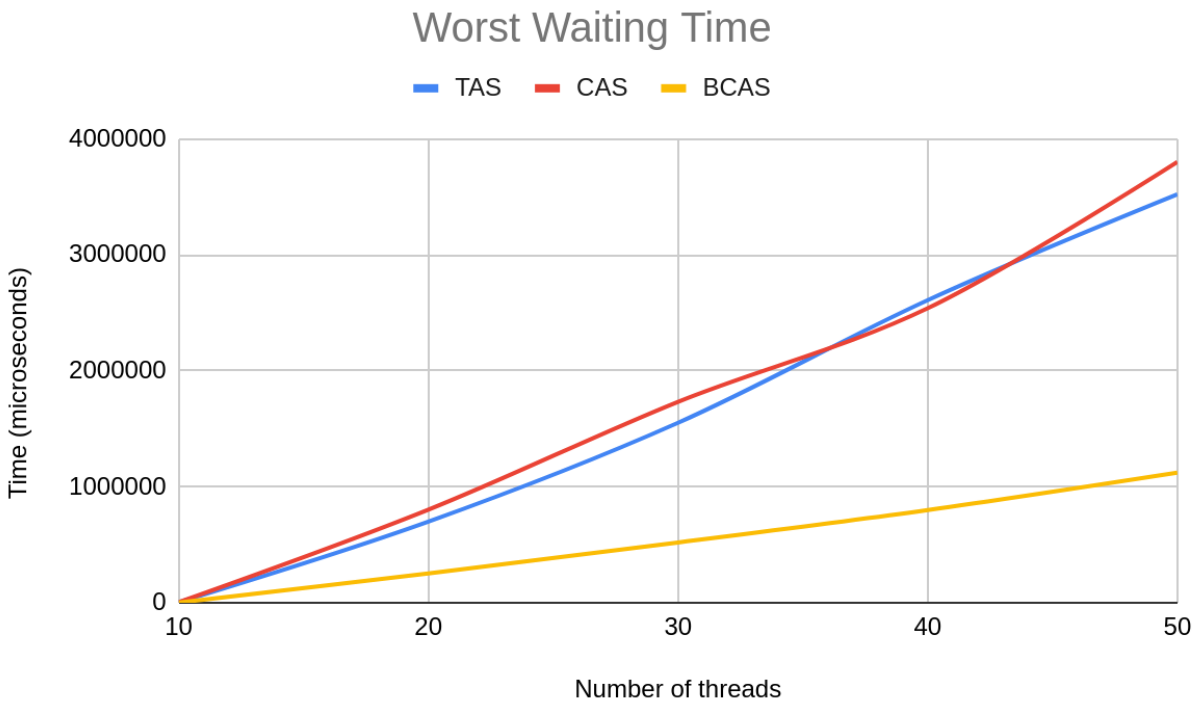
CAS

No. of threads	Time_taken in each iterations (CAS) (microseconds)					Worst Waiting
	1st	2nd	3rd	4th	5th	
10	4627	4497	2669	4205	2720	3743.6
20	768213	592159	920102	680058	1049684	802043.2
30	1580011	1744050	1992618	1628306	1720158	1733028.6
40	2852003	2200015	2492224	2489806	2664052	2539620
50	3652060	3952447	4104106	4075657	3220000	3800854

Bounded CAS

No. of threads	Time_taken in each iterations (Bounded CAS) (microseconds)					Worst Waiting
	1st	2nd	3rd	4th	5th	
10	612	633	566	800	659	654
20	283634	287948	263535	251944	175785	252569.2
30	523972	535945	542544	507934	484039	518886.8
40	843689	799892	755947	799999	791945	798294.4
50	1155999	1115945	1095952	1115944	1120000	1120768

Graph :- (TAS vs CAS vs Bounded CAS)



Analysis : -

- Worst waiting time of TAS and CAS are approximately the same while that of Bounded CAS is different as bounded waiting is satisfied in bounded CAS.
- Average Waiting time of Bounded CAS is more as it has to satisfy bounded waiting so, on average thread has to wait more before entering critical section and after making request to enter critical section while for TAS and CAS average waiting time is almost same.