Report

ES15BTECH11002

Design:

Unique ID is passed to each of the threads.
Localtime function is used to get the time.
usleep function is used to simulate the critical section and remainder section time .
The three algorithms are implemented as per the book.

TAS:
Atomic_flag_test_and_set_explicit is used to set the lock.
Atomic_flag_clear_explicit is used to free the lock

TAS-Bounded:
Waiting queue is implemented as per the book.
The remaining functions are same as that of TAS.

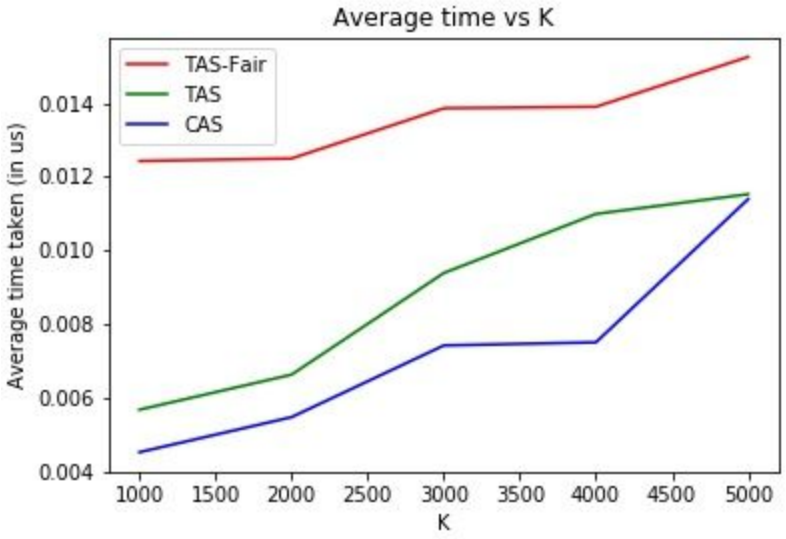
CAS:
Compare_exchange_strong is used for compare and swap functionality.
Expected value and new_val is reset in every iteration of the spin lock, as the above function changes the value of one of the variables every time.

Difficulties:

Output comes jumbled sometimes, can be solved using fprintf() command which is atomic.

Graphs:

Average Time vs K



Analysis:

CAS seems to perform the best among the given algorithms, followed by TAS and TAS-Fair.

Although CAS , requires three parameters to be accesses, it outperforms the remaining 2.