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

## Aim

To Implement Class Mutual Exclusion Algorithm.

## Implementation of Program

In class mutual exclusion algorithm, multiple threads belonging to same session only can enter CS. And if a thread in some session has enter CS, then other thread from other session cant enter the CS. Hence a boolean array active_states of size S (number of session) has been created which is true if thead of that particular session is in CS. Two binary semaphores called mutex and class_mutex have been used. In testCS, a random session is created. Then it is that session is active or not. If it isnt active then it will wait sem_wait(class_mutex) until another threads from active session do sem_post(class_mutex). After that the session is set to active that is active_states[s] = true. If another thread from same session tries to enter CS then it is granted request since the session is active and count is incremented. Mutex semaphore is used for mutual exclusion between count variable. When a thread tries to leave CS, then count is decremented and if count value becomes zero which means all threads in that session have left CS, then sem_post(&class_mutex) is called which allows threads waiting from other sessions to enter CS.

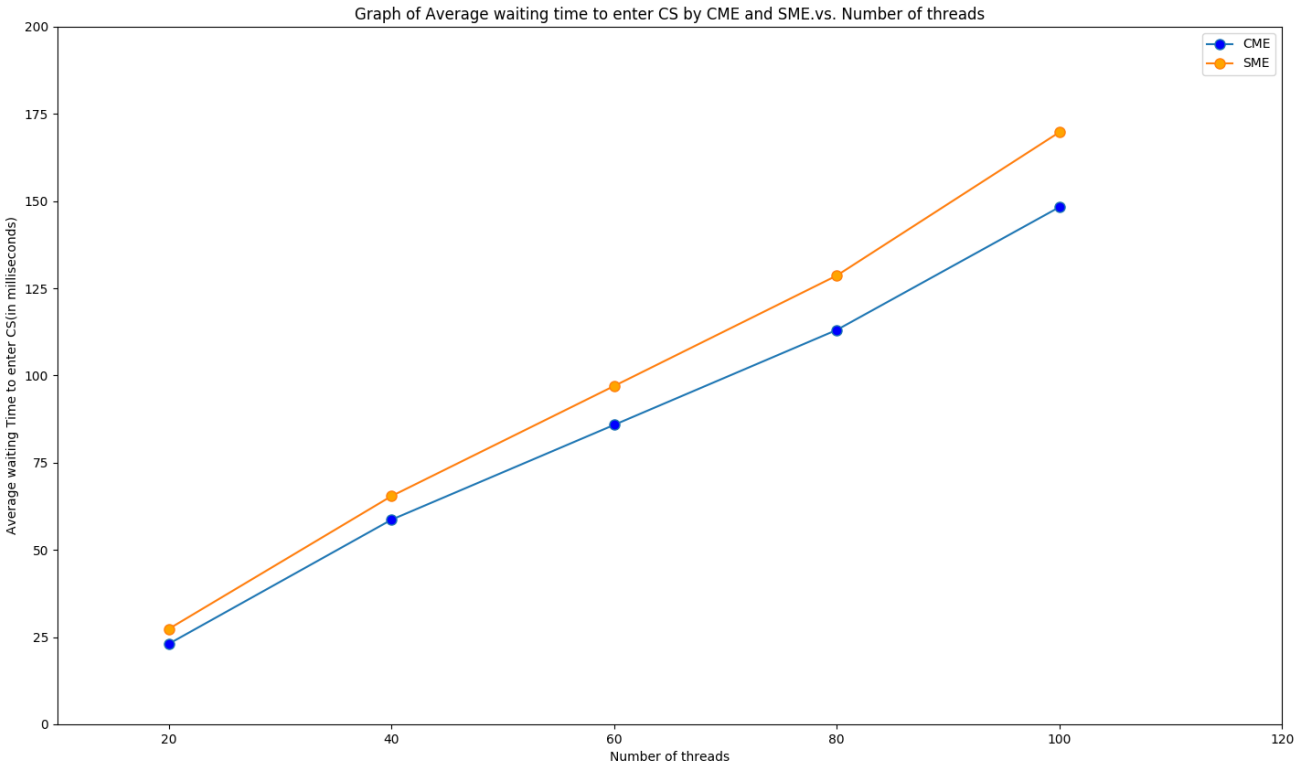
## Comparison of the performance of producer and consumer threads

Input parameters :-
Count of each philosopher thread (k) = 20
Number of sessions (S) = 5
lambda1 = 5
lambda2 = 20
These parameters are kept constant throughout.

n (Number of philosopher threads) have been changed from 20 to 100 in increments of 20.

# Graphs

1) Graph of Average waiting time to enter CS vs Number of threads




# Conclusion

We observe that average waiting time taken by thread to enter CS increases almost linearly as number of threads (n) increases from 20 to 100. We also observe that average waiting time to enter CS by thread is smaller for CME (Class mutual exclusion algo) than for SME (simple mutual exclusion algo). It is obvious since in CME, multiple threads from same session can enter CS. But in SME, only one thread can enter CS and others have to wait. Hence average waiting time to enter CS for CME is less than average waiting time to enter CS for SME.