The program has been designed to implement Class Mutual Exclusion using Conditional Variables.
To ensure CME, the following variations are made which makes it different from ME.

Various identifiers -
running session – holds the integer value of current session .
session_counts – integer array to hold the number of threads of the current session which are running.
waiting_sessions – the number of threads waiting on various sessions.
av_waiting_time – sum of all waiting times.

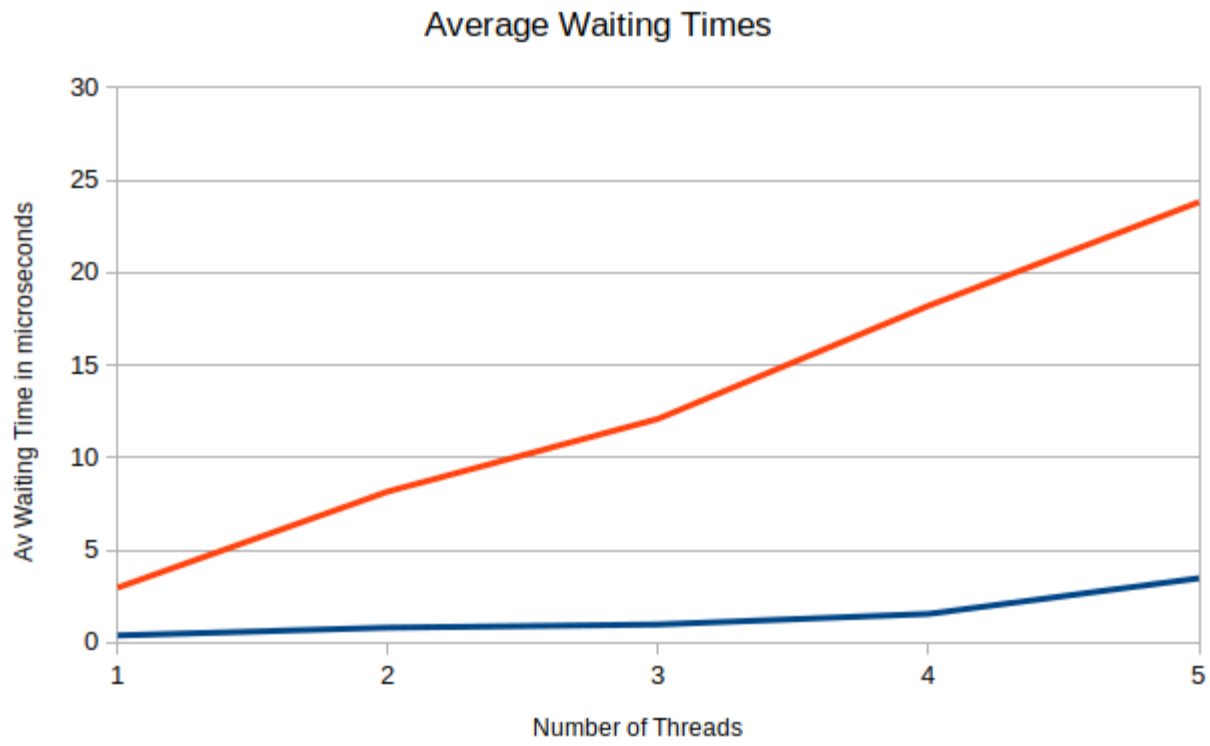
The threads are created and their attributed are initialised. The *func* is the worker function for each thread. Inside, first the session is taken as a random integer. Then the thread writes the request time. Then, it acquires the mutex lock and check various conditions. If there is no running session, then the running session is set to the session of this thread and if the session index of waiting session array has any count of threads > 0, they are all evacuated. The count of session index of session_counts array is increased by 1 for each thread. Else if the running session is not equal to the current session value, the thread is made to wait over the conditional variable cond_var.

The threads which have entered the CS section, writes the time to file, sleep for finite time and then prepare to exit. On exit, first the value of session index of session_counts array is decreased by 1. If the value has become 0, it means that there is no further thread for the current session to run and the cond_var is signalled.
On having signalled, the threads waiting in the else if condition liine 62, will be released. The running session variable is set to the value of current session and all the threads that were waiting on this session are released using the while loop until the value at running_session index of waiting_sessions array becomes 0.

After this, the thread exits the CS and writes the exit time to the file. It then sleeps for some finite time to simulate remainder sectiton.

Graphs -

This implementation is compared to the normal Mutual Exclusion algoritthm (i.e., without the session). The graoh contains 2 curves of waiting time, for each CME and ME. The number of threads are varied from 20 to 100 with intervals of 20. The value of number of iterations is fixed to 20, number of sessions to 100, and the average cs and remainder variable to 5.



The orange line denotes the average waiting time for naive Mutual Exclusion algorithm. The blue line denotes the waiting time for Class Mutual Exclusion.

We can see that the values of waiting times increase the number of threads increase. This should happen because more the number of threads more the threads have to wait to enter in case of mutual exclusion. In case of CME, since many threads can simultaneously implement, it does not increase as steep as naive ME. Also the

important detail is the average waiting time for ME is higher than that of CME. This should be the case since more threads can execute simultaneously in the CME case, which is observed from the log file of CME algorithm.