1.) the data shared among cooperating threads are not correct because there is no way on ensuring th at only one thread may execute inside a critical s ection at a time, hence why we must lock it to mak e sure that all updates to shared data must be exe cuted atomically with respect to other operations on that data.

2.) Because when it is the time frame still allows for that small number of tasks to preempt each ot her and still not have clashes with accessing shar ed data.

3.) Because only one thread is working at a time in the local variables.

4.) It blocks any other calling thread when the mu tex is locked, so that there are no clashes.

5.) With Lock:
real 0m0.187s
user 0m0.222s
sys 0m0.273s

without Lock:

real 0m0.018s
user 0m0.040s
sys 0m0.000s

It differs greatly because with the lock it has to lock the data so that the threads work mutually e xclusively, whereas without the lock they just kee p preempting each other and all the threads can ju st keep on chugging through without waiting for th e other threads to finsih.