Question 6 :

a. Why is threading useful on a single-core processor?

On a single -core processor, threading allows multiple instructions to be processed at a time.

b. Do more threads always mean better performance?

More threads do not always imply a better performance. If we have too many threads for a simple process, it could take more time. Or if a program is not designed to use multiple threads, having many threads will not improve the performance.

c. Is super-linear speedup possible? Explain why or why not.

Super-linear speedup is not possible because even if we have many threads for a program, it could at most have a linear speedup.

1. Why are locks needed in a multi-threaded program?

Locks are needed in a multi-threading to avoid errors in a program. For instance, in a shared-memory, if two threads increment the same variable, at the end it’s not guaranteed that we will have the expected value. To avoid that, we have to use locks to control the access to the variable.

1. Would it make sense to limit the number of threads in a server process?

It makes sense to limit the number of threads in a server process because a processor core can have 2 threads at most normally. If we don’t limit the threads, we could end up with a sever with a bad process performance.