

Mandatory Activity. Concurrent Programming. Lab 09

This activity must be autonomously done by the student. **It must be done prior to the following laboratory class.** It will be used as part of the following laboratory.

Activity

Genetic information is encoded as a sequence of nucleotides: Guanine (G), Adenine (A), Thymine (T), and Cytosine (C). A gene contains an ordered a sequence of nucleotides.

Starting from the vector modulus project, the student must:

- Implement an application that concurrently computes the number of occurrences of a gene in a DNA structure (chromosomes). Examples:
 - o “GT” (gene) in “ABGTA” (chromosome): 1 occurrence.
 - o “AB” (gene) in “ABGAB” (chromosome): 2 occurrences.
 - o “AAA” (gene) in “GTAAAA” (chromosome): 2 occurrences.
- Test the sequential (one thread) and concurrent (multiple threads) implementation using the Visual Studio testing tool.
- Measure context switch. Analyze when the highest benefit is obtained. Identify the number of threads from which the concurrent version is slower than the sequential one.
- Test it with chromosomes of 100,000 nucleotides and genes of 9 nucleotides.
- Invoke `GC.Collect()` and `GC.WaitForFullGCComplete()` after the execution of the algorithm.
- Execute it twice to avoid the effects of JIT compilation. Run it in **Release** mode (not in **Debug** mode).
- In an *Excel* document, show the evolution of execution times relative to the number of threads. Identify the values mentioned in the previous point.

Use the appropriate programming language features learned so far.