Homework Problem 1 | Semir Elezovikj

* Section 1**:** Introduction to Operating Systems
* 1.1: Computer Specifications

I have the following system specifications:

**system\_profiler | grep Processor**

Processor Name: Intel Core i7

Processor Speed: 2.4 GHz

Number of Processors: 1

**system\_profiler | grep Cores**

Total Number of Cores: 4

The main thread creates a new thread t on which it runs a method that repeatedly prints the character "y". Simultaneously, the main thread repeatedly prints the character "x", as shown in Figure 19-1, "Starting a new thread". On a single-processor computer, the operating system must allocate "slices" of time to each thread (typically 20 ms in Windows) to simulate concurrency, resulting in repeated blocks of "x" and "y". On a multiprocessor or multicore machine, the two threads can genuinely execute in parallel, although you still get repeated blocks of "x" and "y" because of subtleties in the mechanism by which Console handles concurrent requests.

The complex answer is ... more complex! The reason that multithreaded programs may often be more efficient than linear ones is because of a hardware "problem". The CPU can execute calculations more quickly than memory and hard storage IO. So, an "add" instruction, for example, executes far more quickly than a "fetch". Caches and dedicated program instruction fetching (not sure of the exact term here) can combat this to some extent, but the speed issue remains.