Many programming languages, especially older ones, provide no language support for concurrency. C and C++ are examples of such languages. Is it essential that a language include syntax for concurrency to be able to write concurrent programs in that language? If not, how is it accomplished? What are the advantages and disadvantages of including support for concurrency in a language?

Concurrency is the ability of different parts or units of a program, algorithm, or problem to be executed out-of-order or in partial order, without affecting the outcome, allowing for parallel execution of the concurrent units, which can significantly improve overall speed of the execution in multi-processor and multi-core systems. In more technical terms, concurrency refers to the decomposability property of a program, algorithm, or problem into order-independent or partially ordered components or units.

Advantages of concurrency:

* Natural model for real-time systems allowing parallel computing for various tasks
* Tasks with longer runtimes do not delay tasks with shorter run times
* Tasks that are waiting for other conditions to be met, can wait, and resume transparently
* Scheduling becomes easy and processes independent of a task do not have to wait for it to finish.
* Calculated in the form of multi-core CPU to maximize the performance gains

Disadvantages of concurrency:

* System hangs if a bad thread occurs, blocking other processes
* Although a process may not directly depend on a waiting process, but it may depend on some other process which is dependent on the waiting process
* Racing condition may occur if the timing of events fails to have proper synchronization
* Since problems are infrequent debugging it can be difficult
* The time slice is the time that the CPU allocates to each thread. Because the time is very short, the CPU constantly switches threads, so that we think that multiple threads are executed at the same time. The time slice is usually tens of milliseconds. Each time you switch, you need to save the current state, so that you can restore the previous state, and this switch is very lossy performance, too often, but cannot play the advantages of multi-threaded programming

References:

Lamport, Leslie (July 1978). "Time, Clocks, and the Ordering of Events in a Distributed System" (PDF). Communications of the ACM.

<https://www.programmersought.com/article/58812938134/>

https://www.programmersought.com/article/4052651281/