

Concurrency, Parallelism, Threads, Processes, Async and Sync — Related?



G. Abhisek

Dec 29, 2018 · 5 min read

Recently, a friend of mine asked me his queries on Concurrency and Parallelism. When I started explaining him his queries, we started discussing other related concepts and nomenclatures such as Threads -> Multi-threaded and Single, Asynchronous and Synchronous. At a point, we were confused with queries like:

How is concurrency related to parallelism?

What is synchronous and asynchronous execution?

What is the importance of synchronous and asynchronous programming in concurrency and parallelism?

How threads fit along with all these concepts?

Many of us sometimes get confused with such queries.

Hold on for a moment and try to answer the above queries and visualize the concepts by yourself.





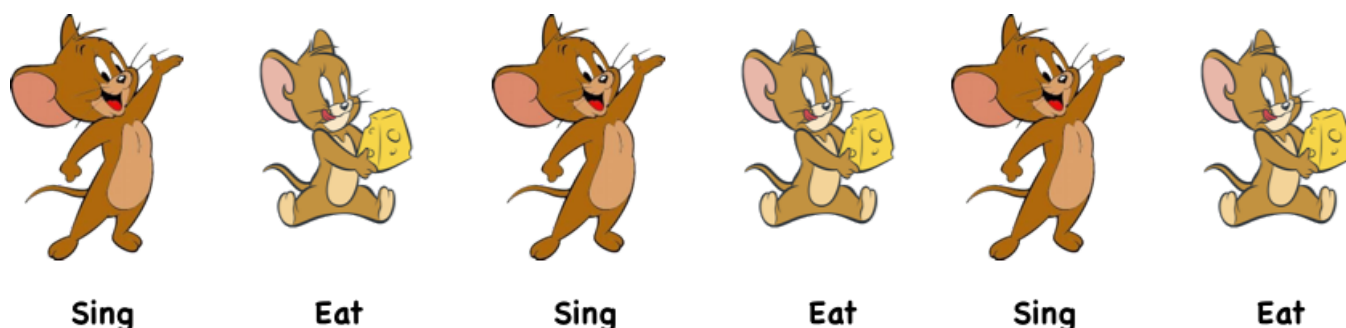
If you are like, "I am confused?" Then we will have an awesome time in this discussion 😊

So let us check whether you have grasped it right. We would analyse and understand what actually they are and their relationship between one another. We will keep our discussion easy and concise.

Concurrency & Parallelism

Concurrency

Consider you are given a task of singing and eating at the same time. At a given instance of time either you would sing or you would eat as in both cases your mouth is involved. So in order to do this, you would eat for some time and then sing and repeat this until your food is finished or song is over. So you performed your tasks concurrently.



You can sing or eat at a time not simultaneously. Don't mind being Jerry 😊

Concurrency means executing multiple tasks at the same time but not necessarily simultaneously. In a concurrent application, two tasks can start, run, and complete in overlapping time periods i.e Task-2 can start even before Task-1 gets completed.

In the computer science world, the way how concurrency is achieved in various processors is different. In a single core environment (i.e your processor is having a single core), concurrency is achieved via a process called context-switching. If its a multi-core environment, concurrency can be achieved through parallelism.



Execution of tasks in a single core environment. Tasks are context switched between one another.

Parallelism

Consider you are given two tasks of cooking and speaking to your friend over the phone. You could do these two things simultaneously. You could cook as well as speak over the phone. Now you are doing your tasks parallelly.

Parallelism means performing two or more tasks simultaneously. Parallel computing in computer science refers to the process of performing multiple calculations simultaneously.



Task - 1



Task - 2

Two tasks are being performed simultaneously over the same time period.

How is concurrency related to parallelism?

- Concurrency and Parallelism refer to computer architectures which focus on how our tasks or computations are performed.
- *In a single core environment, concurrency happens* with tasks executing over same time period via context switching i.e at a particular time period, only a single task gets executed.
- *In a multi-core environment, concurrency can be achieved via parallelism* in which multiple tasks are executed simultaneously.

Threads & Processes

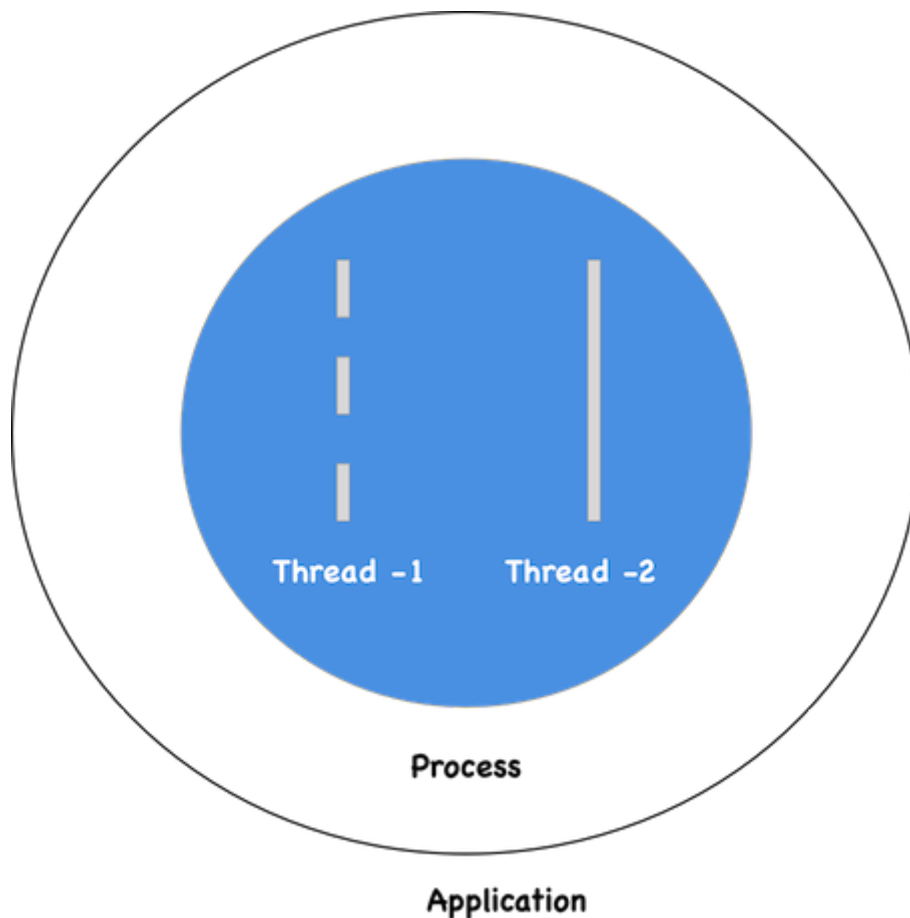
Threads

Threads are a sequence of execution of code which can be executed independently of one another. *It is the smallest unit of tasks that can be executed by an OS. A program can be single threaded or multi-threaded.*

Process

A process is an instance of a running program. *A program can have multiple processes. A process usually starts with a single thread i.e a primary thread but later*

down the line of execution it can create multiple threads.



Distribution of Processes and Threads in an application.

Synchronous and Asynchronous

Synchronous

Imagine you were given to write two letters one to your mom and another to your best friend. You can not at the same time write two letters unless you are a pro ambidextrous.

In a synchronous programming model, tasks are executed one after another. Each task waits for any previous task to complete and then gets executed.

Asynchronous

Imagine you were given to make a sandwich and wash your clothes in a washing machine. You could put your clothes in the washing machine and without waiting for it to be done, you could go and make the sandwich. Here you performed these two tasks asynchronously.

In an asynchronous programming model, when one task gets executed, you could switch to a different task without waiting for the previous to get completed.

Synchronous and Asynchronous in a single and multi-threaded environment

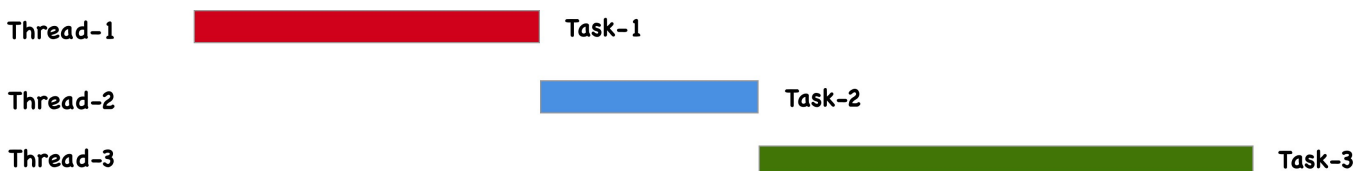
Synchronous

Single Threaded:



Each task gets executed one after another. Each task waits for its previous task to get executed.

Multi-Threaded:



Tasks get executed in different threads but wait for any other executing tasks on any other thread.

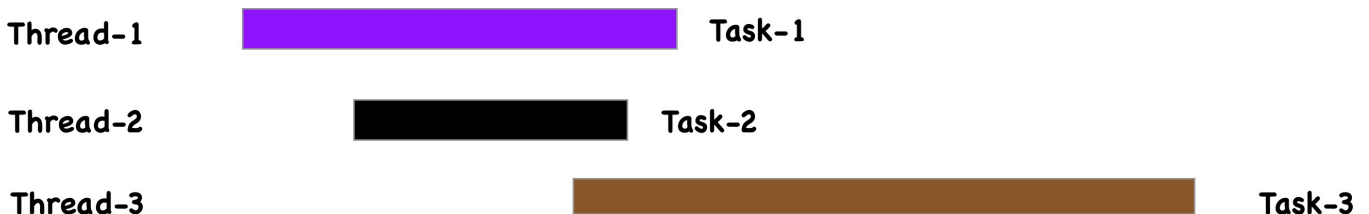
Asynchronous

Single Threaded:



Tasks start executing without waiting for a different task to finish. At a given time a single task gets executed.

Multi-Threaded:



Tasks get executed in different threads without waiting for any tasks and independently finish off their executions.

What is the role of synchronous and asynchronous programming in concurrency and parallelism?

- Asynchronous programming model helps us to achieve concurrency.
- Asynchronous programming model in a multi-threaded environment is a way to achieve parallelism.

Summing up

Concurrency and Parallelism ->

Way tasks are executed.

Synchronous and Asynchronous ->

Programming model.

Single Threaded and Multi-Threaded ->

The environment of task execution.

References

- <https://softwareengineering.stackexchange.com/questions/190719/the-difference-between-concurrent-and-parallel-execution>
- <https://stackoverflow.com/questions/748175/asynchronous-vs-synchronous-execution-what-does-it-really-mean>
- <https://codewala.net/2015/07/29/concurrency-vs-multi-threading-vs-asynchronous-programming-explained/>
- https://youtu.be/cN_DpYBzKso
- <https://medium.com/flawless-app-stories/basics-of-parallel-programming-with-swift-93fee8425287>

I would love to hear from you

You can reach me for any query, feedback or just want to have a discussion by the following channels:

Twitter — @G_ABHISEK

LinkedIn

StackOverflow

GMail: abhisekbunty94@gmail.com

In order to get connected instantly

SkypeId — gabhisekbunty

Please feel free to share with your fellow developers.

[Programming](#) [Computer Science](#) [iOS](#) [Computer Architecture](#) [Concurrency](#)

[About](#) [Help](#) [Legal](#)