A **multi-core processor** is a [microprocessor](https://en.wikipedia.org/wiki/Microprocessor) on a single [integrated circuit](https://en.wikipedia.org/wiki/Integrated_circuit) with two or more separate [processing units](https://en.wikipedia.org/wiki/Central_processing_unit), called cores, each of which reads and executes [program instructions](https://en.wikipedia.org/wiki/Instruction_set).[[1]](https://en.wikipedia.org/wiki/Multi-core_processor#cite_note-1) The instructions are ordinary [CPU instructions](https://en.wikipedia.org/wiki/Instruction_set) (such as add, move data, and branch) but the single processor can run instructions on separate cores at the same time, increasing overall speed for programs that support [multithreading](https://en.wikipedia.org/wiki/Multithreading_(computer_architecture)) or other [parallel computing](https://en.wikipedia.org/wiki/Parallel_computing) techniques.

There are intercore communication methods

In [computer architecture](https://en.wikipedia.org/wiki/Computer_architecture), **multithreading** is the ability of a [central processing unit](https://en.wikipedia.org/wiki/Central_processing_unit) (CPU) (or a single core in a [multi-core processor](https://en.wikipedia.org/wiki/Multi-core_processor)) to provide multiple [threads of execution](https://en.wikipedia.org/wiki/Thread_(computer_science)) concurrently, supported by the [operating system](https://en.wikipedia.org/wiki/Operating_system). This approach differs from [multiprocessing](https://en.wikipedia.org/wiki/Multiprocessing). In a multithreaded application, the threads share the resources of a single or multiple cores, which include the computing units, the [CPU caches](https://en.wikipedia.org/wiki/CPU_cache), and the [translation lookaside buffer](https://en.wikipedia.org/wiki/Translation_lookaside_buffer) (TLB).

 multithreading aims to increase utilization of a single core

How its done ?

Program is broken into multiple small threads .

Multithreading is the ability of a Central Processing Unit (CPU) to break a single process into multiple threads of execution and run them concurrently.

Processes are memory intensive operations. To help them run faster and use memory more efficiently, programmers will break large programs into smaller tasks called execution threads.

An execution thread is the smallest sequence of programmed instructions that can be independently managed by a scheduler, the abstraction responsible for determining when, where, and in what order threads are allowed to execute.

Each thread…

* Shares memory and data with other threads running within the same parent process
* Can communicate with other threads with few to no system calls
* Is lightweight consuming less time and resources for creation, execution, and context switching

In programming, a thread maintains a list of information relevant to its execution, including the priority schedule, exception handlers, a set of CPU registers, and stack state in the address space of its hosting process.

pre-emptive multithreading (controlled by OS)in which the available processor time is shared. All threads get an equal time slice and are serviced in a queue-based model. During thread switching, the context of a pre-empted thread is stored and reloaded in the next thread in the queue. This takes so little time, that the running threads seem to execute in parallel.

Then there’s cooperative multithreading, in which context switching is controlled by the thread

Concurrent processing means that only one thread will be handled at a time, but the system will create efficiencies by moving quickly between two or more threads.

Parallel processing means the system is actually handling more than one thread at a given time.

to the human user, either parallel or concurrent process, or processes that are mixed, are all experienced as parallelism in real-time.

 an instruction stream is called a [thread](https://www.techopedia.com/definition/27857/thread-operating-systems) and the instance of the computer program that is executing is called a process. Each process has its own memory space where it stores threads and other data the process requires to execute.

While multithreading allows a process to create more threads to improve responsiveness, multiprocessing simply adds more CPUs to increase speed.

* Concurrency: To run a group of tasks concurrently means you don’t have to wait for one task to complete before starting another. They may execute simultaneously, take turns progressing, or some combination of those two states. To the end user they may as well have occurred at the same time.
* Parallelism: A specific type of concurrency where tasks are truly executed simultaneously. This feat is only possible in multi-core environments