

Параллельное и конкурентное исполнение программ

В этом уроке поговорим про три парадигмы программирования.

Parallel, параллельные вычисления или параллельность. Одновременно обрабатываемые взаимосвязанные подпрограммы.

Например: нужно решить одну и ту же задачу с различными вариантами входных параметров и получить значения. Мы можем спроектировать нашу программу так, чтобы она не решала задачу за задачей, а просчитывала несколько задач одновременно.

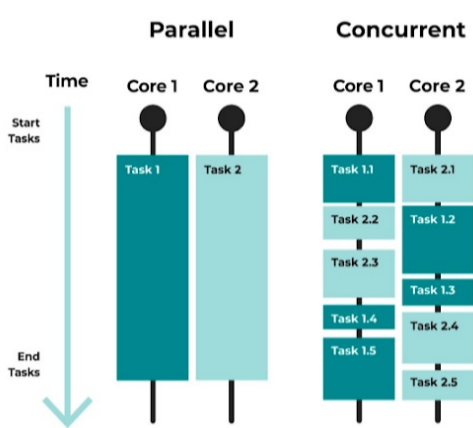
[Статья на wiki](#)

Многопоточность. Это способность центрального процессора CPU — или одного ядра в многоядерном процессоре — одновременно выполнять несколько процессов или потоков, соответствующим образом поддерживаемых операционной системой.

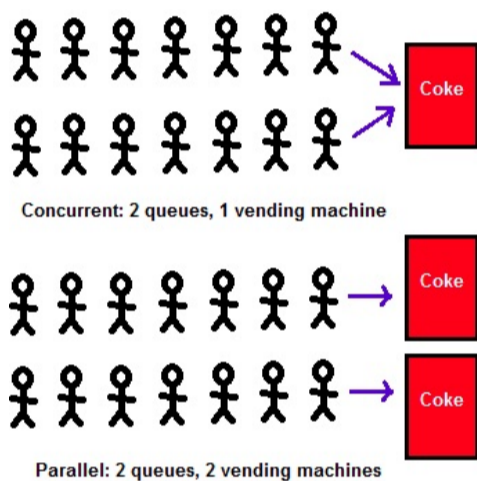
Многопоточность хорошо применима при операциях с CPU. Однако далее вы узнаете про такое явление на Питоне, как GIL. Оно не позволяет эффективно применить многопоточность.

Concurrency, или конкурентность. Иногда еще называют «Одновременность». Но в действительности при конкурентности все операции выполняются *вместе*, но *не одновременно*. Именно этот подход реализован в библиотеке `asyncio`.

На диаграмме можно посмотреть, чем отличается параллельность от конкурентности:



Объяснение на примере из реальной жизни:



Возможные реализации этих парадигм программирования:

- треды
- процессы
- корoutines