

Python- разработчик с нуля

Урок 012 — мультипоточность

Цели и задачи урока

Цель

a. Узнать, как можно ускорить выполнение программ.

Или нет

Задачи

- a. Модель параллельных вычислений
- b. Потoki
- c. Блокировки
- d. Очереди
- e. Процессный параллелизм
- f. Практика

Модель параллельных вычислений

- Сердце компьютера — процессор. Он дает энергию процессам (задачам), чтобы те выполнялись.



- **Многозадачность** — это когда процессор быстро переключается между процессами (задачами) и нам кажется, что они выполняются одновременно.
- **Многопоточность** — это когда в одном процессе (задаче) есть несколько потоков выполнения кода.
- В таком стиле все становится **асинхронным** — нельзя заранее сказать, когда и где выполнится конкретный участок кода.

Потоки

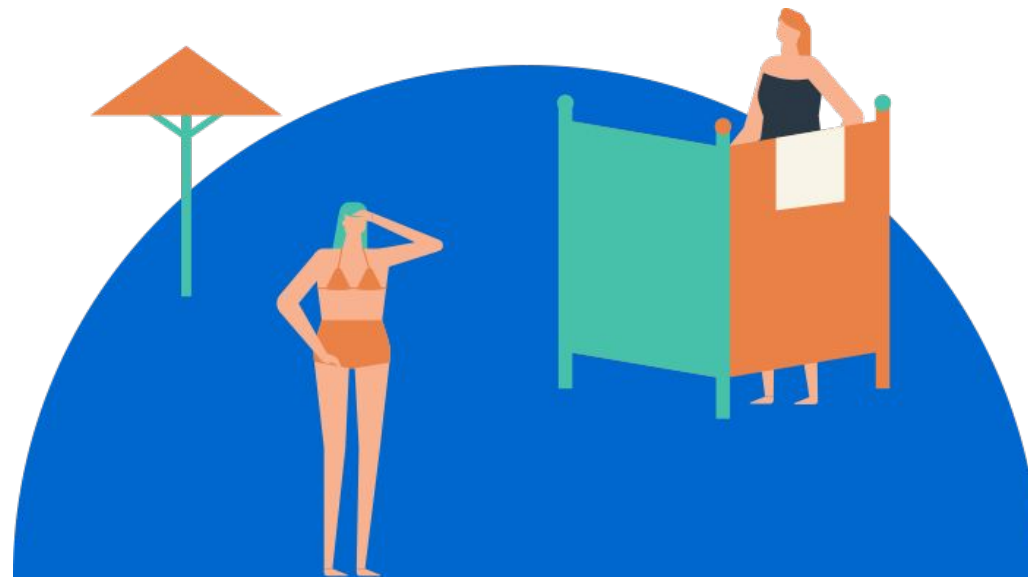
Skillbox

- Модуль threading дает возможность создать потоки внутри процесса, выполняемого Python.
- Потоки похожи на спагетти



Блокировки

- Когда потокам надо изменять одни и те же данные, то нужна блокировка.
- Блокировку можно представить как кабинку: если один вошел, то другие ждут.



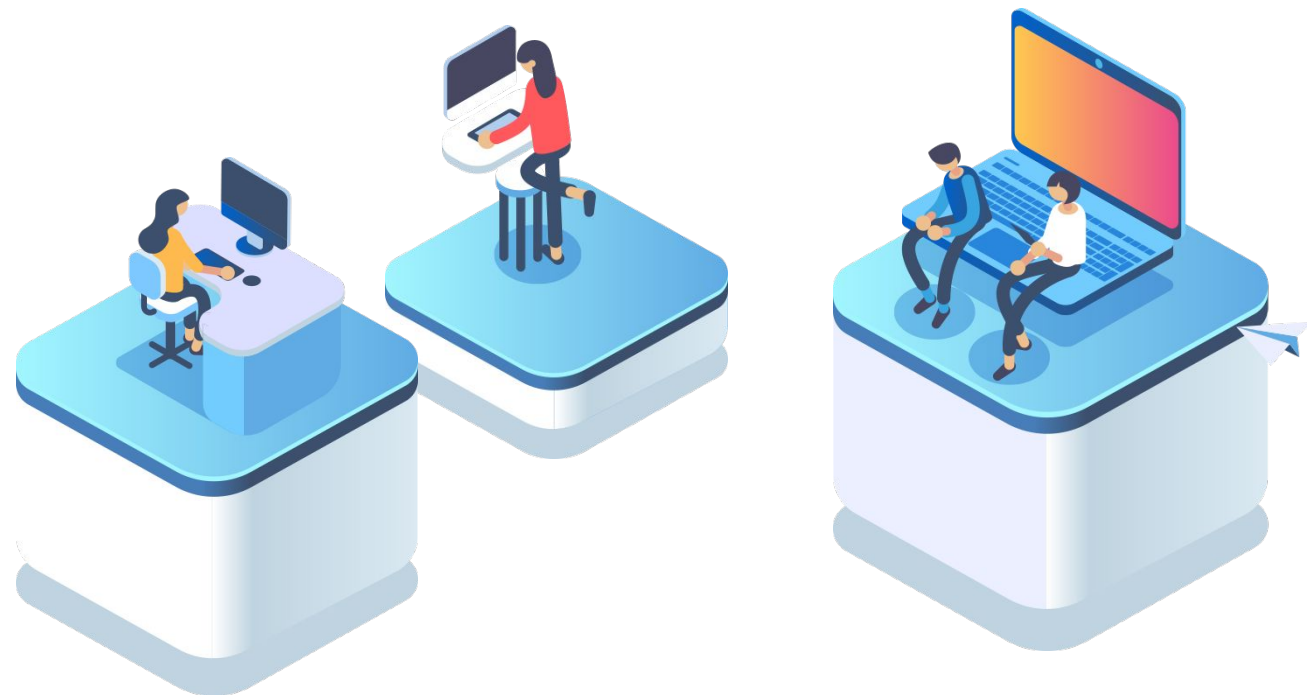
Очереди

- Еще один способ обмениваться данными — очередь.
- Одни потоки кладут в очередь, другие достают.
- Как мячики в спортлото — из хаотичного набора выбираются несколько, и вот они уже обрабатываются последовательно.



Процессный параллелизм

- Иногда лучше создать несколько процессов-обработчиков
- Тогда можно полностью использовать все ядра всех процессоров компьютера.



Практика

- Асинхронное программирование обычно используется для работы с внешними ресурсами и обработки запросов пользователей.
- Напишем скрипт, который вычисляет объем первых страниц сайтов из списка.



Домашнее задание

Напишите скрипт, который ходит по сайту и находит неверные ссылки.

Итак, мы изучили

- Распараллеливание выполнения кода позволяет достичь большей производительности. Но это не точно.
- Мы узнали как многопоточно и многозадачно программировать на python
- Увидели нюансы многопоточности и научились их обходить
- Программировать в асинхронном стиле сложно, но можно.