



Занятие №24

Многопоточное,
асинхронное и
мультипроцессорное
программирование. GIL





1. В чём проблема процессов?

В чём проблема процессов?

У современных процессоров больше одного ядра, поэтому мы можем ускорить выполнение задач, используя модуль многопроцессной обработки multiprocessing. Задачи, завязанные на процессоре – это программы, которые большую часть времени своей работы выполняют вычисления на процессоре (тривиальные математические вычисления, обработку изображений и т.д.). Если вычисления могут выполняться независимо друг от друга, мы имеем возможность разделить их между доступными ядрами процессора, тем самым получив значительный прирост в скорости обработки.

В чём проблема процессов?

В задачах, связанных с вводом-выводом, multiprocessing также может повысить производительность, но издержки, как правило, оказываются выше, чем при использовании multithreading.

Вы можете заметить, что нагрузка на центральный процессор будет выше, когда вы будете использовать multiprocessing по сравнению с обычным циклом for или даже multithreading. Так происходит, потому что ваша программа использует не одно ядро, а несколько. И это хорошо!

Помните, что multiprocessing имеет свои издержки на управление несколькими процессами, которые обычно серьезнее, чем издержки multithreading. (Multiprocessing порождает отдельные интерпретаторы и назначает каждому процессу свою область памяти, так что да!) То есть, как правило, лучше использовать облегченную версию multithreading, когда вы хотите выкрутиться таким способом (вспомните про задачи, связанные с вводом-выводом). А вот когда вычисление на процессоре становится бутылочным горлышком, приходит время модуля multiprocessing. Но помните, что с большой силой приходит большая ответственность.

Если вы породите больше процессов, чем ваш процессор может обработать в единицу времени, то заметите, что производительность начнет падать. Так происходит, потому что операционная система должна делать больше работы, тасуя процессы между ядрами процессора, потому что процессов больше.

