

assert
unit test

Лабораторная работа 6

Поток - абстрактно уровню ОС, содержащая отдельный контекст (стек) выполнения, позволяющая выполнять несколько задач одновременно или переключаться между ними.

* Потоки содержатся внутри процесса и совместно разделяют некоторые ресурсы.

Процесс - экземпляр программы во время выполнения, независимый объект, которому выделены время и ресурсы системы.

Многopроцессорность - модуль в Python, позволяющий достичь параллелизма, напоминающий по функциональности поток, но использующий другие инструменты для синхронизации процессов и обмена данными: Queue (очереди) и Pipe (каналы).

* Модуль threading предлагает упрощение

работы с потоками и позволяет программировать запуск нескольких операций одновременно.

* Multiprocessing предоставляет Pool - вариант API и Process.

Демон (процесс) - завершают свою работу если завершился родительский процесс, для них запрещено создавать самостоятельно дочерние процессы.

GIL - механизм блокировки, когда интерпретатор Python запускает в работу только один поток за раз.

Принцип работы GIL:

- 1) потоки удерживают его, пока выполняются
- 2) освобождают его при блокировании для операций ввода / вывода
- 3) если таких нет, интерпретатор периодически проводит проверку

4) при проверке в главном потоке запускаются обработчики сигналов

5) GIL отключается и включается вновь при кратком освобождении GIL другие потоки имеют шанс на запуск.

* Threading используют для операций ввода-вывода: загрузка ресурсов из интернета или чтение файлов и папок на компьютере.

* Multiprocessing используют для работы с трудоемкими операциями (циклы, обработка больших количество объектов, парсинг сложных форматов, математические вычисления)

* Threading и multiprocessing не стоит использовать если придется часто синхронизировать данные, блокируя другие потоки.

* За основной идеей синхронизации потоков при помощи блокировки стоит сохранение логики алгоритма изменения данных,

которые должны изменяться в один момент.

- * В Python реализации синхронизации потоков при помощи блокировки состоит в использовании `with self.mutex`, обеспечивающего выполнение части кода в одном потоке.

- * Структура данных очереди имеет в качестве идеи, что код выполняемой программы становится более простым.

- * Отличие очередей от очередей с приоритетом в том, что в обычной очереди принцип «первый вошел — первый вышел».

В очереди с приоритетом элемент с самым высоким приоритетом уходит из нее в первую очередь.