

|           | Достоинства  | Недостатки   |
|-----------|--|--|
| Threading | Запуск нескольких потоков в программе рационализирует использование процессора, поскольку время простоя процессора уменьшается.  | Многопоточность затрудняет достижение производительности с точки зрения скорости вычислений в однопроцессорной системе по сравнению с производительностью в многопроцессорной системе.   |
|           | Все потоки процесса имеют доступ к его глобальным переменным. Если глобальная переменная изменяется в одном потоке, она видна и другим потокам. Поток также может иметь свои собственные локальные переменные. | Все потоки в программе совместно используют одни и те же данные, следовательно, всегда существует проблема безопасности, потому что любой неизвестный поток может изменить данные.   |
|           | Позволяет программе оставаться отзывчивой, пока один поток ожидает ввода, а другой одновременно запускает графический интерфейс.   | Требуется синхронизация, чтобы избежать взаимного исключения. Это приводит к увеличению использования памяти и процессора.   |
|           | Повышает скорость вычислений, поскольку каждое ядро или процессор обрабатывает отдельные потоки одновременно.  | Не прерывается/ не уничтожается  |
|           |  | Даже хотя некий процесс может содержать множество потоков, отдельная недопустимая операция в пределах одного потока может отрицательно сказываться на имеющейся обработке всех прочих потоков в этом процессе и может вызывать в результате крушение всей программы целиком.   |
|           |  | Разделение одних и тех же ресурсов может выступать неким преимуществом, для совместного использования ресурсов также требуется аккуратное рассмотрение подробностей с тем, чтобы совместные данные вычислялись правильно и их обработка была корректной. Интуитивно непонятные проблемы, которые могут быть вызваны небрежной координацией потоков включают в свой состав взаимные блокировки, зависания и состояние конкуренции |

|                   |  |  |
|-------------------|--|--|
| <b>Processing</b> | Отдельное пространство памяти у каждого процесса (безопасность)  | Увеличение объема памяти   |
|                   | Предотвращает ограничения GIL(Python Global Interpreter Lock — это своеобразная блокировка, позволяющая только одному потоку управлять интерпретатором Python) для cPython   | Операционная система защищает процессы от взаимного влияния. Процессы, нуждающиеся в обмене данными, должны явно организовать такой обмен посредством механизмов межпроцессного взаимодействия |
|                   | Процессы дочерних процессов можно прервать/уничтожить  | Передача информации между процессами происходит медленнее, чем между потоками, т.к. несколько процессов не могут использовать одно пространство памяти   |
|                   | Сетевые механизмы хорошо приспособлены к IPC (межпроцессное взаимодействие), поскольку они работают с процессами, выполняющимися на различных узлах сети, а не на одном и том же узле.                             | Процессы более энергозатратны, чем потоки, потому что открытие/закрытие процессов занимает больше времени  |
|                   | Модуль multiprocessing предоставляет механизмы, пригодные для осуществления IPC по сети.   |  |
|                   | Нагрузка распределяется между несколькими процессорами, что повышает надежность. Если один процессор выходит из строя, его отказ может немного замедлить скорость системы, но система будет работать бесперебойно. |  |