# Как ThreadPool распределяет задачи по таскам?

Есть приоритеты, планировщик сам передает свободным потокам задачи.

# Как ThreadPool распределяет задачи по таскам?

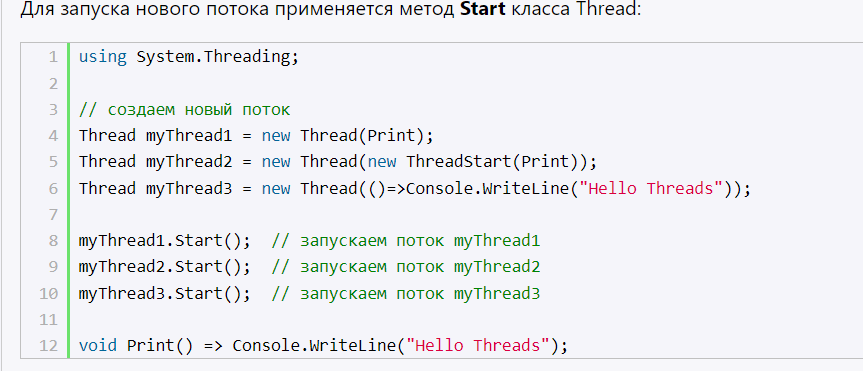
Из очереди задач, ThreadPool выберет одну из них в соответствии с **внутренним алгоритмом планирования**.

**Параллельное выполнение**: Если в пуле есть несколько свободных потоков, задачи могут выполняться параллельно. ThreadPool автоматически управляет количеством потоков в зависимости от текущей загрузки системы и настроек пула.

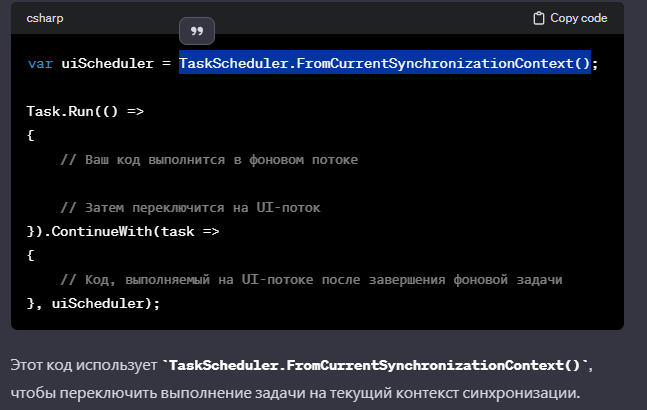
# Как привязать выполнение таски к конкретному потоку?

мне кажется если создать новый поток, то таска будет привязана к нему

Либо через TaskScheduler – ThreadPool.UnsafeBackGroundQueueWorkItem



Или так но это для WPF и не безопасно



# Зачем запускать таски в отдельных потоках?

<https://code-maze.com/csharp-new-thread>

Это позволяет отдельно выполнить задачу, без прерываний. Например веб браузер грузит файл в отдельном потоке и в это же время можно менять громкость музыки

# Что такое .ConfigureAwait(false)

**ConfigureAwait (false)** следует ли продолжать выполнение асинхронной задачи в том же контексте синхронизации (SynchronizationContext), из которого была запущена задача.

Обычно, если код запущен из UI-потока (например, в приложении WPF или WinForms), результат асинхронной операции будет возвращаться в тот же UI-поток, что может быть важно для обновления пользовательского интерфейса.

Однако, в некоторых случаях, например, в библиотеках или фоновых задачах, может потребоваться избежать возврата в исходный контекст синхронизации. Использование ConfigureAwait(false) позволяет предотвратить возврат в оригинальный контекст синхронизации, что может улучшить производительность и избежать нежелательных взаимодействий с UI-потоком