

.net каждый поток представлен объектом класса Thread. Чтобы организовать свой поток, необходимо создать объект этого класса. Конструктору может передаваться в качестве параметра делегат, инкапсулирующий метод, выполняемый в потоке. Доступно 2 типа делегатов: второй позволяет при запуске метода передать ему параметр-объект:

```
public Thread(ThreadStart start);  
public Thread(ThreadStart start, int maxStackSize);  
public delegate void ThreadStart();  
public delegate void ParameterizedThreadStart(object obj);
```

Дополнительный параметр конструктора может использоваться для указания максимального размера стека, выделяемого потоку. Для запуска потока необходимо вызвать метода Start() (перегруженная версия принимает объект, передаваемый методу потока как параметр).

Основные свойства Thread: CurrentThread, Name, ManagedThreadId (уникальный числовой идентификатор управляемого потока), ThreadState, IsAlive, Priority (ThreadPriority: Lowest, BelowNormal, Normal, AboveNormal, Highest), IsBackground.

Thread содержит методы: Suspend() (приостановка), Resume() (возобновление), Sleep() (приостановление, можно указать кол-во миллисек или TimeSpan), Yield() (передача управления след. ожидающему потоку), Join() (дождаться завершения работы данного потока или блокировка его выполнения на опр. время). Завершение работы выбранного потока выполняет метод Abort(). Данный метод генерирует исключение ThreadAbortException, которое можно подавить catch блоком и может быть отслежено потоком, который кто-то пытается уничтожить, а метода ResetAbort() отклоняет запрос на уничтожение потока.

.net поддерживает также специальный механизм, называемый пул потоков, который поддерживает такие технологии как асинхронные делегаты, таймеры, asp.net.

Для работы с пулом используется класс ThreadPool. Метода SetMaxThreads() позволяет изменить емкость пула (по умолчанию 25*число процессорных ядер), а SetMinThreads() устанавливает минимальное число потоков (по умолчанию = числу процессорных ядер). Для помещения метода в пул потоков используется метод QueueUserWorkItem(). Он принимает делегат типа WaitCallback и, возможно, параметр инкапсулируемого метода.

В пространстве имен System определен атрибут [ThreadStatic], применяемый к статическим полям. Если поле помечено таким атрибутом, то каждый поток будет содержать свой экземпляр поля. Для [ThreadStatic]-полей не рекомендуется делать инициализацию при объявлении, т.к. инициализация выполняется только в одном потоке.

Синхронизация потоков - координирование действий потоков для получения предсказуемого результата. Существует несколько категорий средств синхронизации потоков:

приостановка выполнения потока (Suspend(), Resume(), Sleep(), Yield(), Join());

1. блокирующие конструкции;
2. конструкции подачи сигналов;
3. незадерживающие средства синхронизации.

Блокирующие конструкции обеспечивают исключительный доступ к ресурсу (поле, фрагмент кода). Для организации .net предоставляет классы Monitor, Mutex, Semaphore, SemaphoreSlim, а язык C# оператор lock (критические секции).

lock(<объект синхронизации>) { <операторы критической секции> }

lock - скрытый способ работать со статическим классом Monitor, который содержит методы Enter(), Exit(), Wait() - применяется внутри критической секции и снимает с нее блокировку.

Для реализации незадерживающей синхронизации, используется статический класс System.Threading.Interlocked.