

A decorative graphic on the left side of the slide. It consists of a blue parallelogram and a light green parallelogram, both tilted at an angle. The blue shape is in the foreground, and the green shape is partially behind it. They are set against a dark blue background with faint, lighter blue diagonal stripes.

МНОГОНИШКОВО програмиране



Planning



Execution

Multi threading




Нишка

- **Многонишкова (multithreaded) програма** - програма, която може да изпълняват едновременно няколко редици от програмни инструкции.
- **Нишка (thread)** - редица от програмни инструкции
- Нишките са част от процеса и споделят неговите ресурси
- Изпълнението на нишките напомня на изпълнението на няколко програми едновременно



Нишки, но защо?

- Позволяват изпълнението на няколко операции едновременно
- Максимизират полезната употреба на ресурсите на процесора (позволява на програмата да използва всичките ядра)
- Намалява времето за изпълнение (не е нужно да чакаме една операция да приключи, за да започне друга)



Нишки, но как?

- В C# трябва да включим библиотеката за нишки:

```
using System.Threading;
```

- Създава се обект за новата нишка. Като параметър към конструктора се подава референция към метода, който ще се изпълнява от нишката:

```
Thread thread = new Thread(threadMethod);
```

- Изпълнението на нишката трябва да бъде стартирано, чрез извикването на `.Start()` метода:

```
thread.Start();
```



Видове нишки в C#

- Нишка на главния метод (main thread) - нишката, с която се стартира и изпълнява стандартно програмата
- Нишка на преден план (foreground thread) - нишка, която остава активна до приключване на работата си, дори когато главната нишка е приключила вече с изпълнението си. Thread обектът по подразбиране е foreground thread.
- Нишка на зададен план (нишка във фонов режим, background thread) - нишка, която прекратява работата си най-късно, когато и главната нишка я прекрати. За да направите една нишка във фонов режим, трябва да зададете свойство на обекта на нишката:

```
thread.IsBackground = true;
```



Алтернативно създаване на нишка

Нишка може да бъде създадена и за изпълнението на метод от друг обект:

```
public class ExampleThread {  
    public void print()  
    {  
        for (int i = 0; i < 3; i++) {  
            Console.WriteLine("Other Thread");  
        }  
    }  
}
```




Алтернативно създаване на нишка

В main метода или друг метод създаваме нишката:

```
// Създаваме обект от ExampleThread класа
ExampleThread obj = new ExampleThread();


Thread thread = new Thread(new ThreadStart(obj.print));
thread.Start();
```

Създаване на нишка с подаване на параметри към метода

Какво ако искаме да използваме параметри в метода, който ще се изпълнява на отделна нишка?

- Ламбда израз
- Делегати
- `ParameterizedThreadStart`



Създаване на нишка с подаване на параметри към метода

- Пример с ламбда израз:

```
//нека методът изглежда така:  
public void MyMethod(string param1, int param2)  
{  
    //do stuff  
}
```

```
//можем да стартираме нишката по следния начин:  
Thread thread = new Thread(() => MyMethod("param1",5));  
thread.Start();
```



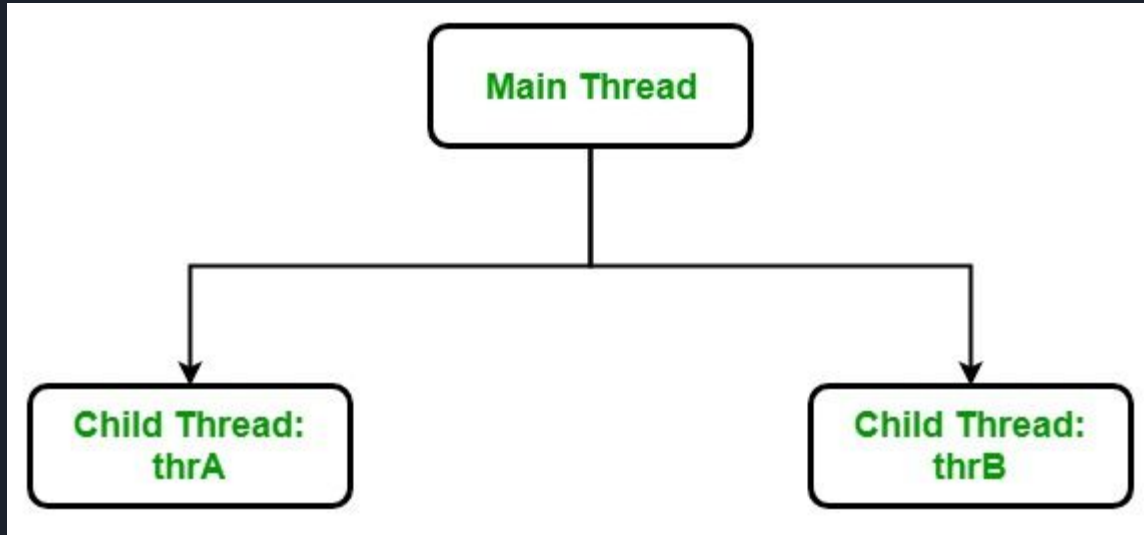
Главна нишка

При стартиране на програмата, тя веднага стартира и една нишка - т.нар. **Главна нишка**.

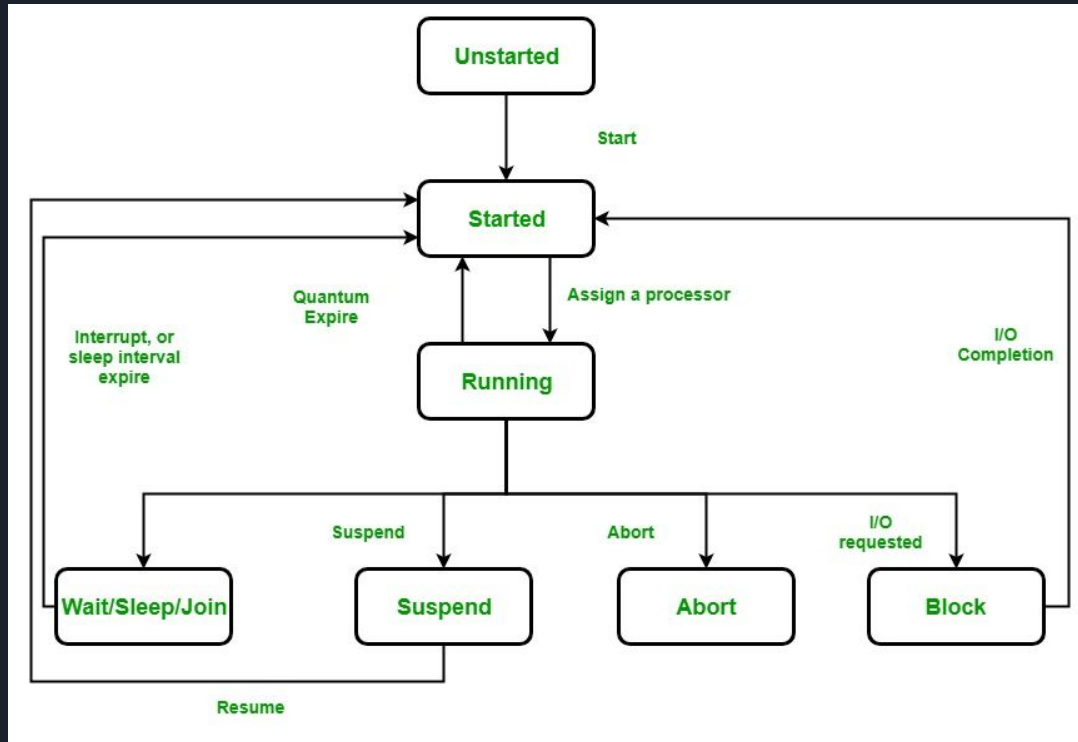
Важни свойства:

- Останалите нишки се падат „деца“ (child threads) на главната нишка
- Често, но не винаги, тази нишка е нишката, която последна трябва да приключи със своето изпълнение, понеже в нея могат да се изпълняват важни действия свързани с приключването на програмата

Йерархия на нишките



Жизнен цикъл на нишките






Жизнен цикъл на нишките (2)

Нишките в който и да е момент от изпълнението на програмата имат някое от следните състояния:

1. Незапочната (Unstarted)
2. Подлежаща на изпълнение (Runnable)
3. Изпълняваща се (Running)
4. Не се изпълнява (Not Running)
5. Мъртва (Dead)



Нишка в незапочнато (unstarted) състояние

Незапочнато състояние: когато е създаден обект от Thread класа, той се намира в незапочнато състояние, което ще рече, че нишката още не е започнала изпълнението си. С прости думи: не е извикан Start() метода.




Подлежаща на изпълнение (Runnable)

Състояние на подлежаща на изпълнение нишка: Нишка, която е готова да се изпълни се прехвърля в Runnable състояние. В това състояние, нишката дори е възможно да се изпълнява или е възможно да се изпълни във всеки един момент. Отговорност на диспечера на нишките (thread scheduler) е да разреши на нишката да се изпълни. С други думи това е моментът след извикването на Start() метода.



Състояние на изпълнение (Running)

Нишката в това състояние се изпълнява. С други думи, нишката се обработва от процесора.



Състояние на НЕизпълнение (Not Running)

Състоянието на НЕизпълнение се предизвиква при някое от следните събития:

- Извикване на Sleep() метода
- Извикване на Wait() метода
- Изпълнение на I/O (входно/изходна) заявка
- Изпълнение на Suspend() метода

За справка погледнете диаграмата от 13 слайд.



Мъртво състояние (Dead state)

Нишката е изпълнила задачата си и вече не е нужна.
Тогава тя навлиза в мъртво, прекратено, състояние.

Благодаря за вниманието!
Автор: Петър Р. Петров,
ПГЕЕ „К. Фотинов“, гр. Бургас

