

XII. Párhuzamos programvégrehajtás

A feladatok gyorsabb, hatékonyabb elvégzésének érdekében az utóbbi években speciális lehetőségként jelent meg a párhuzamos programvégrehajtás. Gondolhatunk a többfeladatos operációs rendszerekre, vagyis arra például, hogy miközben a programunkat fejlesztjük, és valamilyen szövegszerkesztőt használunk, a produktívabb munkavégzés érdekében, gondolkodásunkat serkentendő, kedvenc zenénket hallgathatjuk számítógépünk segítségével.

A párhuzamos programvégrehajtás programjaink szintjén azt jelenti, hogy az operációs rendszer felé több végrehajtási feladatot tudunk definiálni.

Például egy adatgyűjtési feladatban az egyik szál az adatok gyűjtését végzi, a másik szál pedig a korábban begyűjtött adatokat dolgozza fel.

A .NET keretrendszer, ahogy több más fejlesztőeszköz is, lehetőséget ad arra, hogy egy program több végrehajtási szálát definiáljon. Ezekről a végrehajtási szálakról az irodalomban, online dokumentációban *threads*, *threading* néven olvashatunk.

XII.1. Szálak definiálása

Általában elmondhatjuk, hogy a párhuzamos végrehajtás során az alábbi lépéseket kell megtenni (ezek a lépések nem csak ebben a C# környezetben jellemzőek):

- Definiálunk egy függvényt, amelynek egy paramétere lehet és a visszatérési típusa *void*. Ez több rendszerben kötelezően *run* névre hallgat.
- Ennek a függvénynek a segítségével egy függvénytípust, delegáltat definiálunk. Könyvtári szolgáltatásként a *ThreadStart* ilyen, használhatjuk ezt is, ahogy a következő példa is mutatja. Egy paraméteres delegált a *ParameterizedThreadStart*.
- Az így kapott delegáltat felhasználva készítünk egy *Thread* objektumot.
- Meghívjuk az így kapott objektum *Start* függvényét.

A párhuzamos végrehajtás támogatását biztosító könyvtári osztályok a *System.Threading* névtérben találhatóak.

Ezek után az első példaprogram a következőképpen nézhet ki:

Példa:

```
using System;  
using System.Threading;
```

```

class program
{
    public static void szal()
    {
        Console.WriteLine("Ez a szálprogram!");
    }
    public static void szal1(object o)
    {
        Console.WriteLine("A kapott paraméter: {0}!",o);
    }
    public static void Main()
    {
        Console.WriteLine("A főprogram itt indul!");
        ThreadStart szalmutato=new ThreadStart(program.szal);
        // létrehoztuk a fonal függvénytmutatót, ami a
        // szal() függvényre mutat
        Thread fonal=new Thread(szalmutato);
        // létrehoztuk a párhuzamos végrehajtást végző objektumot
        // paraméterül kapta azt a delegáltat (függvényt)amit
        // majd végre kell hajtani
        fonal.Start();
        // elindítottuk a fonalat, valójában a szal() függvényt
        // a fonal végrehajtása akkor fejeződik be amikor
        // a szal függvényhívás befejeződik
        //
        Thread fonal1=new Thread(program.szal1);
        //a ThreadStart vagy ParameterizedThreadStart közvetlen
        //elhagyható
        fonal1.Start(10);
        Console.WriteLine("Program vége!");
    }
}

```

XII.2. Szálak vezérlése

Ahogy az előző példában is láthattuk, egy szál végrehajtását a **Start** függvény hívásával indíthatjuk el, és amikor a függvény végrehajtása befejeződik, a szál végrehajtása is véget ér. Természetesen a szál végrehajtása független a **Main** főprogramtól.