In .NET 4.0, we have a set of new API's to simplify the process of adding parallelism and concurrency to applications. This set of API's is called the "Task Parallel Library (TPL)" and is located in the System.Threading and System.Threading.Tasks namespaces.

The Parallel class found in the System.Threading.Tasks namespace “provides library-based data parallel replacements for common operations such as for loops, for each loops, and execution of a set of statements”

using System.Threading;

using System.Threading.Tasks;

C#

static void GenerateNumbers()

{

for (int i = 0; i < 10; i++)

{

Console.WriteLine("Method1 - Number: {0}", i);

Thread.Sleep(1000);

}

}

static void PrintCharacters()

{

string str = "dotnetcurry";

for (int i = 0; i < str.Length; i++)

{

Console.WriteLine("Method2 - Character: {0}", str[i]);

Thread.Sleep(1000);

}

}

static void PrintArray()

{

int[] arr = {1, 2, 3, 4, 5, 6, 7, 8 };

foreach (int i in arr)

{

Console.WriteLine("Method3 - Array: {0}", i);

Thread.Sleep(1000);

}

}

In order to call these 3 methods concurrently, we will use Parallel.Invoke() as shown below:

C#

static void Main(string[] args)

{

Parallel.Invoke(

new Action(GenerateNumbers),

new Action(PrintCharacters),

new Action(PrintArray)

);

Console.ReadLine();

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading;

using System.Threading.Tasks;

namespace \_01\_console

{

class Program

{

static void GenerateNumbers()

{

for (int i = 0; i < 10; i++)

{

Console.WriteLine("Method1 - Number: {0}", i);

Thread.Sleep(1000);

}

}

static void PrintCharacters()

{

string str = "dotnetcurry";

for (int i = 0; i < str.Length; i++)

{

Console.WriteLine("Method2 - Character: {0}", str[i]);

Thread.Sleep(1000);

}

}

static void PrintArray()

{

int[] arr = { 1, 2, 3, 4, 5, 6, 7, 8 };

foreach (int i in arr)

{

Console.WriteLine("Method3 - Array: {0}", i);

Thread.Sleep(1000);

}

}

static void Main(string[] args)

{

Parallel.Invoke(

new Action(GenerateNumbers),

new Action(PrintCharacters),

new Action(PrintArray)

);

Console.ReadLine();

}

}

}