Parallel, Asynchronous and Multithreaded Applications

Course code: GOC407

The course is intended for experienced developers who would like to improve their knowledge and want to learn about parallel and asynchronous applications. There will learn the basics of threads and objects and familiarize themselves with the products which enable this kind of programming. A lot of these new features come with .NET Framework 4 (e.g. Task Parallel Library) and 4.5 (e.g.async/await). We will talk about the differences and find solutions for dealing with IO bound and CPU bound problems. We will also look at Power Threading Library. In the last few lessons of the course we will concentrate on the solutions dealing with the synchronization such as locky, mutexes, lock-free operations and we will also talk about their specifice and the suitability to particular solutions.

What we teach you

To create applications which will use all power of HW and therefore they are quicker

Applications which are scaling well on new devices

To think about the basic problems and system sources in a new innovated way

Required skills

Experience with creation of the application in C# languag

Course outline

Parallelism
- Current state, future of command
- Description of typical problems

Synchronization
- Lock/Monitor
- Spin-wait
- Interlocked - lock-free operations
- Mutex, EventWaitHandle, Semaphore, ReaderWriterLockSlim
- Assembling of locks and their types
- Lock free thinking
- A little bit of .NET memory model

IO bound operations
- Description
- Specifications
- Asynchronous (non-blocking) processing

CPU bound operations
- Description
- Specifications
- Support in .NET
- Class Thread, in other words all hard-boiled
- ThreadPool

NET 4
- PLINQ
- Task Parallel Library
- Improved ThreadPool
- Work stealing

NET 4.5
- Asynchronous (non-blocking) processing

GOPAS®

- New key words async/await
- New "Async" methods in .NET
- Cooperation with Task/Task
- Progress reporting

Power Threading Library

- AsyncEnumerator
- Classes which can enable you to work in multithreaded environment and to scale (ReaderWriterGate, spin-locky...)

GOPAS®