

# Jazyk C++ – vlákna a paralelné programovanie v C++

Kód kurzu: CPP\_THR

Kurz je určený pre pokročilých vývojárov, ktorí sa chcú zoznámiť s novými konceptmi a triedami pre tvorbu prenositeľných viacvláknových aplikácií pomocou štandardných C++ 11 knižníc. Vysvetlíme typické problémy paralelných aplikácií ako napr. súbeh operácií na zdieľaných dátach ("Race Condition") alebo uviaznutie ("Deadlock"). Predstavíme vlákna ("Threads") a použitie paralelných úloh ("Task Parallelism"). Pozrieme sa aj na kritické sekcie dôležité pre vzájomné vylúčenie vlákien, ich použitie pri objektoch typu "Monitor" a na odovzdávanie správ, výhody neblokujúcich operácií a atomické dátové typy.

## Pre koho je kurz určený

Kurz je určený pre pokročilých vývojárov, ktorí sa chcú zoznámiť s novými konceptmi a triedami pre tvorbu prenositeľných viacvláknových aplikácií pomocou štandardných C++ 11 knižníc.

## Čo vás naučíme

- Používať triedy a funkcie z moderných C++ 11 viacvláknových knižníc.
- Vytvárať aplikácie schopné využiť súčasný masívne paralelný hardware.
- Premýšľať o základných problémoch návrhu paralelného software.

## Požadované vstupné znalosti

- Skúsenosť s programovaním v jazyku C++.
- Triedy, konštruktory, deštruktory, automatická správa zdrojov.
- Veľkou výhodou sú predchádzajúce skúsenosti s tvorbou viacvláknových aplikácií pomocou iných knižníc (Posix, WinAPI, MFC, ...).

## Osnova kurzu

- Vlákna
- Vytváranie vlákien pomocou triedy Thread
- Fork and Join model
- Samostatné vlákna
- Odovzdávanie parametrov
- Transport výnimiek
- Paralelné úlohy
- Vytváranie paralelných úloh pomocou Async
- Návrátová hodnota paralelnej úlohy, trieda Future
- Čakanie na výsledok paralelnej úlohy
- Trieda Packaged\_task
- Trieda Promise a detaily triedy Future
- Trieda Shared\_future.
- Kritické sekcie
- Trieda Mutex
- Vzťah "Happens-before", zamykanie a odomykanie.
- Trieda Lock\_guard a Unique\_lock
- Monitor objekty
- Jednorázová inicializácia pomocou Call\_once.
- Signalizovanie pomocou Condition Variables
- Producent a konzument
- Metódy Notify a Wait
- Falošné prebudenie (Spurious Wakeup)
- Protokol v súvislosti so zmenou stavu zdieľanej premennej
- Atomické typy

### GOPAS Praha

Kodaňská 1441/46  
101 00 Praha 10  
Tel.: +420 234 064 900-3  
[info@gopas.cz](mailto:info@gopas.cz)

### GOPAS Brno

Nové sady 996/25  
602 00 Brno  
Tel.: +420 542 422 111  
[info@gopas.cz](mailto:info@gopas.cz)

### GOPAS Bratislava

Dr. Vladimíra Clementisa 10  
Bratislava, 821 02  
Tel.: +421 248 282 701-2  
[info@gopas.sk](mailto:info@gopas.sk)



Copyright © 2020 GOPAS, a.s.,  
All rights reserved

# Jazyk C++ – vlákna a paralelné programovanie v C++

- Trieda Atomic
- Blokujúce a neblokujúce operácie
- Lock-free dátové štruktúry

**GOPAS Praha**  
Kodaňská 1441/46  
101 00 Praha 10  
Tel.: +420 234 064 900-3  
[info@gopas.cz](mailto:info@gopas.cz)

**GOPAS Brno**  
Nové sady 996/25  
602 00 Brno  
Tel.: +420 542 422 111  
[info@gopas.cz](mailto:info@gopas.cz)

**GOPAS Bratislava**  
Dr. Vladimíra Clementisa 10  
Bratislava, 821 02  
Tel.: +421 248 282 701-2  
[info@gopas.sk](mailto:info@gopas.sk)



Copyright © 2020 GOPAS, a.s.,  
All rights reserved