

C30

Jazyk C++ - vlákna a paralelní programování v C++

Popis:

Kurz Jazyk C++ - Vlákna a paralelní programování v C++ je určen pro pokročilé vývojáře, kteří se chtějí seznámit s novými koncepty a třídami pro tvorbu přenositelných vícevláknových aplikací s pomocí standardních C++ 11 knihoven. Vysvětlíme typické problémy paralelních aplikací jako souběh operací na sdílených datech ("race condition") nebo uváznutí ("deadlock"). Představíme vlákna ("threads") a použití paralelních úloh ("task parallelism"). Uvidíme kritické sekce důležité pro vzájemné vyloučení vláken a jejich použití u objektů typu "monitor". Podíváme se na předávání zpráv, výhody neblokujících operací a atomické datové typy.

Absolvent kurzu bude umět:

- Řešit problémy paralelních aplikací jako souběh operací na sdílených datech ("race condition") nebo uváznutí ("deadlock")
- Užívat vlákna ("threads") a paralelní úlohy ("task parallelism")
- Rozeznat kritické sekce důležité pro vzájemné vyloučení vláken a jejich použití u objektů typu "monitor"
- Předávat zprávy, výhody neblokujících operací a atomické datové typy.

Požadavky pro absolvování kurzu:

Zkušenosti s programováním.

Kurz určen pro:

Vývojáře, kteří se chtějí seznámit s novými koncepty a třídami pro tvorbu přenositelných vícevláknových aplikací s pomocí standardních C++ 11 knihoven.

Literatura:

Všichni účastníci školení obdrží materiály společnosti OKsystem.

Technické vybavení:

Všechny učebny jsou vybaveny nadstandardními počítači připojenými k Internetu, učebny jsou prostorné, klimatizované, bezbariérové a s připojením na Wi-Fi. V případě zájmu lze školení

absolvovat online live.

Osnova:

- Vytváření vláken pomocí třídy thread
- Fork and join model
- Samostatná vlákna
- Předávání parametrů
- Transport výjimek
- Vytváření paralelních úloh pomocí async
- Návratová hodnota paralelní úlohy, třída future
- Čekání na výsledek paralelní úlohy
- Třída packaged_task
- Třída promise a detaily třídy future
- Třída shared_future
- Třída mutex
- Vztah "happens-before", zamykání a odemykání
- Třída lock_guard a unique_lock
- Monitor objekty
- Jednorázová inicializace pomocí call_once
- Producent a konzument. Metody notify a wait
- Falešné probuzení (spurious wakeup)
- Protokol kolem změny stavu sdílené proměnné
- Třída atomic
- Blokující a neblokující operace
- Lock-free datové struktury