

Linux Multithread-/Multicore-Programmierung - Live-Online-Training

Ziele - Ihr Nutzen

Dieses Training zeigt, wie die Prozessorleistung von Linux-Systemen - insbesondere bei modernen Multicore-Architekturen - bestmöglich genutzt werden kann.

Er richtet sich an Softwareentwickler, die neue Applikationen entwickeln oder existierende Anwendungen optimieren müssen, und eignet sich insbesondere auch für die Anwendung unter Embedded-Linux.

Schwerpunkt des Trainings ist die Vorstellung des Pthread-APIs der NPTL-Bibliothek (Native Posix Threads Library).

Dabei wird nicht nur der Funktionsumfang der Threadbibliothek eingehend behandelt, sondern auch auf mögliche Probleme und Lösungen sowie auf Richtlinien für einen guten Softwareentwurf eingegangen.

Darüber hinaus zeigt das Training aber auch, um wie viel einfacher eine Multithread-Applikationsentwicklung mithilfe von OpenMP sein kann, und bietet somit einen Einblick in eine völlig andere Art der Softwareentwicklung für Multithreading-Systeme.

Teilnehmer

Software-Entwickler, Software-Architekten, Projektleiter

Voraussetzungen

Kenntnis der Programmiersprache C.

Live Online Training

* Preis je Teilnehmer, in Euro zzgl. USt.

Anmeldecode: L-LIN-MMP

Präsenz-Training - Deutsch

Dauer

3 Tage

Linux Multithread-/Multicore-Programmierung - Live-Online-Training

Inhalt

Einführung

- Prinzipielle Funktionsweise eines Threads
- Zuordnung von Code und Daten
- Threads vs. Prozesse
- Der Posix Standard und das Pthread API
- Das NPTL-Thread-Modell
- Schedulingkonzepte
- Threadzustände

Threads

- Prioritäten und Scheduling
- Timesharing-Threads
- Realtime-Threads

Synchronisation

- Mutex
- Rekursiver Ressourcenzugriff
- Fairnessverhalten
- Deadlocks
- Prioritätsinversion
- Prioritätsvererbung
- Ceiling-Priorität

Multicore-Programmierung

- Wann lohnt sich Multicore?
- Amdahl'sches Gesetz
- Ansätze zur Leistungsoptimierung
- False Sharing
- Prozessor-Affinität
- Spinlocks
- Richtlinien für einen guten Applikationsentwurf

OpenMP

- Was ist OpenMP?
- Beispiel: Multithreadprogrammierung ohne (explizite) Threads

Praktische Übungen

- Entwicklung von Anwendungsprogrammen zur Analyse des Laufzeitverhaltens von Timesharing- und Realtime-Threads unter verschiedenen Bedingungen
- Entwicklung einer Multithread-Applikation, die vom Benutzer zu beliebigen Zeiten zuverlässig beendet werden kann.
- Aufspüren der Multithread-Probleme einer vorgegebenen Anwendung, sowie deren Behebung mithilfe geeigneter Synchronisationsmechanismen
- Parallelisierung einer sequenziellen Applikation mit OpenMP
- Laufzeitmessungen und Effizienzanalysen