

Multitask-/Multithread-Programmierung in Embedded Systemen

Karl Nieratschker, SKT Nieratschker

Abstract:

Die stetig steigenden Anforderungen an heutige Embedded Systeme führen immer häufiger zum Einsatz von Multitasking- bzw. Multithreading-Systemen (MT-Systeme). Applikationsentwickler/innen, denen die dafür notwendige Erfahrung noch fehlt, sind oft versucht, sich diese empirisch zu erarbeiten. Diese Vorgehensweise ist allerdings gefährlich, denn viele MT-basierte Probleme sind von äußerst subtiler Natur und machen sich oft erst bemerkbar, wenn das Produkt bereits ausgeliefert ist. Der Zweck dieses Seminars ist es, ein solides Allgemeinverständnis für den Umgang mit MT-Systemen zu vermitteln. Es zeigt, wie MT-Systeme generell funktionieren, welche Arten es gibt und wodurch sie sich prinzipiell unterscheiden. Anhand von konkreten C- und C++-Beispielen wird dargestellt, worauf es bei der MT-Programmierung - insbesondere aus dem Blickwinkel der Embedded-Programmierung - ankommt, und was für die Stabilität und Effizienz einer MT-Applikation wichtig ist.

Inhalt:

- Prozessor-/Systemarchitektur, Singlecore-/Multicore-/AMP-/SMP-Systeme
- Tasks/Threads/Prozesse, Schedulingmodelle
- Threadzustände, Ausführung von Funktionen und Methoden durch Threads
- Threaderzeugung und -beendigung, Parameterübergabe, Rückgabe von Ergebnissen
- Speichermodell: Probleme durch Compiler- und Prozessoroptimierungen
- Ablaufsteuerung: Abfragen von Zustandsinformationen, Arbeiten mit Events
- Zugriffssteuerung: Race Conditions, wechselseitiger Ausschluss, Deadlocks, Prioritätsinversion, atomare Operationen, Interrupt Service Routinen
- Multicore-Programmierung: Kosten/Nutzen-Betrachtungen, negative Auswirkungen von Hardwareoptimierungen, typische Probleme beim Umstieg von Singlecore- auf Multicore-Systeme, Hinweise für gutes Multithread-Design

Nutzen:

Ziel dieses Seminar ist es, Teilnehmer/innen, die in diesem Bereich noch keine oder nur wenig Erfahrung haben, ein prinzipielles Verständnis für die Arbeitsweise von MT-Systemen, die Art und Weise der parallelen Programmierung, sowie ein Gefühl für die Art der Fehler zu vermitteln, die dabei auftreten können. Anhand von konkreten Beispielen zeigt der Vortrag, auf welche Weise die vorgestellten Konzepte in verschiedenen Systemumgebungen realisiert sein können. Anhand der Gemeinsamkeiten und Unterschiede der Implementierungen zeigt sich, was das Wesentliche ist und worauf es weniger ankommt. Der generelle Nutzen für die Teilnehmer/innen besteht darin, die Konzepte leicht auf konkrete Anwendungen übertragen zu können. Dies ist z.B. nützlich, wenn es darum geht, die Eignung eines Betriebssystems für ein neues Produkt zu beurteilen, oder um sich schnell in ein neues Betriebssystem einzuarbeiten zu können.

Autor:

Basierend auf einem Studium der technischen Informatik an der TU München verfügt Karl Nieratschker über eine mehr als 30-jährige Erfahrung im Bereich der Embedded-Programmierung und betriebssystemnahen Softwareentwicklung. Als selbständiger Trainer, Softwareberater und Coach unterstützt er seit 1999 Kunden bei der Einführung neuer Softwaretechnologien. Besondere Schwerpunkte dabei sind Objektorientierte Programmierung in Embedded Systemen, sowie Multithreading- und Multicore-Programmierung.

Kontakt:

Internet: www.skt-nieratschker.com
Email: office@skt-nieratschker.com