



Übersicht

Realzeitsysteme sind Systeme, die neben den funktionalen Anforderungen auch noch zeitlichen Anforderungen genügen.

Typischerweise liegen die **zeitlichen Anforderungen** im Sekundenbereich und darunter (Milli-/Mikrosekunden)

Typische Fragestellungen

- **Top-Aktuelles Thema**
- **Sehr viel Entwicklung und Veränderung in den letzten Jahren**
 - Früher: Viele (unterschiedliche) Aufgaben in eine Singlethread-Lösung
 - Modern: Viele (sequentielle) Aufgaben in eine Multithread-Lösung
 - Linux als [Standardbetriebssystem mit Realzeiteigenschaften](#).

Typische Fragestellungen

- Wieviel „Zeit“ liegt zwischen dem 4. Juni 1969, 4:03 Uhr und 13 Sekunden und dem 28. September 2012, 18:45 Uhr und 54 Sekunden?
- Welchen Einfluss hat der Ort auf die Zeit?
- Welchen Zeitbereich kann man in einer Integer-Variablen ausdrücken?
- Mit welcher zeitlichen Genauigkeit arbeitet Windows, mit welcher Linux?
- Wie kann man programmtechnisch Zeiten erfassen?
- Haben zwei Systeme die gleiche Zeit?

Einsatzbereiche



Herausforderungen

- Unterschiedliche Darstellungsformen für Zeiten.
- Rechnungen mit Zeiten sind sehr komplex.
- Durch Parallelität gibt es zu schützende, kritische Abschnitte.
- System, Algorithmus und Implementierung haben Einfluss auf das Zeitverhalten.
- Zeitsynchronisation zwischen verteilten Systemen.
- Die Verarbeitungsleistung moderner Rechner ist nicht konstant.

Fehlerhafte Realzeitsysteme

Prominente Beispiele

- Zune-Player von Microsoft
 - Modelle der ersten Generation stürzten am 31. Dezember 2008 ab und ließen sich nicht neu starten. Der Hersteller gab ein Problem mit dem Schaltjahr 2008 als Grund für die Fehlfunktion an.
- Mars-Pathfinder (NASA)
 - Fehlendes Protokoll zur Behandlung der Prioritätsinversion ließ Pathfinder für mehrere Tage komplett ausfallen.

Organisation

Vorlesung / Übung

- **Dienstag**, 11:30 – 13:00 Uhr
 - Termine in Moodle

Labortermine

- **Mittwochs**, 09:45 – 11:45 Uhr
 - Termine in Moodle

Organisation (Fortsetzung)

Praktikum

- 2er Gruppen
- Hardware: Raspberry Pi Board
- Projekt RoboCar: E/A Anbindungen mit Realzeitaspekt
 - Sensorik: z.B. Ultraschall Sensoren Auswertung
 - WLAN Anbindung, Steuerung der E/A mittels CLI
 - Teams reporten im 2 Wochen Rhythmus
 - Mittwochs ab 9:45 Uhr im F035
- Präsentation
- Diskussion
- Demo

Organisation (Fortsetzung)

Prüfung

- mündlich
- Zur Vorbereitung: Aufgabensammlung siehe Moodle
- Inhalt:
 - Vorlesung
 - Übung und
 - Praktikum.

Übung

- Rechnen von (gegebenen) Aufgaben.
 - zuerst selbständiges Lösen der Aufgabe (mit persönlicher Hilfestellung).
 - danach gemeinsames Lösen der Aufgabe (unter Berücksichtigung der erkannten Probleme).
- Aufgaben sind teilweise alte Prüfungsaufgaben.
- Vermittlung von Lernstoff
 - Zeitbegriffe
 - Posix API
 - Zuverlässigkeit und Security
 - Realzeitnachweis

Moderne Realzeitsysteme kompakt

Das Buch zur Vorlesung

- Jürgen Quade und Michael Mächtel
- Dpunkt-Verlag, 274 Seiten
1. Auflage: Okt. 2012 ISBN: 978-3-89864-830-1 Preis 32.90 Euro
- Mind. 80 Prozent des Buches bilden den Stoff der Vorlesung.
- Einige Exemplare stehen in der Bibliothek zur Ausleihe zur Verfügung.



Vorlesungsinhalte (Überblick)

- Einführung
- Realzeitbetrieb
- Systemsoftware
- Realzeitprogrammierung
- Realzeitarchitekturen
- Safety (Betriebssicherheit)
- Formale Beschreibungsmethoden
- Realzeitnachweis

Realzeitbetrieb

Zentrale Beschreibungsgrößen

- Techn. Prozess, „Außenwelt“
- Rechenprozess, „Taskset“
- Systemsoftware

Systemaspekte

- Unterbrechbarkeit
- Prioritäten
- Ressourcen

Realzeitbedingungen

- Auslastungsbedingung
- Rechtzeitigkeitsbedingung
- Harte und weiche Realzeit

Systemsoftware

Firmware

Realzeitbetriebssysteme

- Systemcalls
- Taskmanagement
- Memory Management
- I/O-Management
- Timekeeping

Realzeitprogrammierung

Tasks

Schutz kritischer Abschnitte

- Semaphore
- Prioritätsinversion
- Unterbrechungsmodell

Inter-Prozess- Kommunikation

- Events
- Signals
- Peripheriezugriff
- Bitoperationen

Umgang mit Zeiten

- Zeiterfassung
- Zeitvergleich
- Differenzzeitmessung
- Schlafenlegen

Realzeitarchitekturen

- Ohne spezielle Systemsoftware
- Basierend auf Standard- OS
- Threaded Interrupts
- Userland-to-Kernel
- RTOS
- Multicore
- Multikernel-Architektur

Safety (Betriebssicherheit)

- Grundbegriffe
- Berechnung der Verfügbarkeit einzelner Komponenten
- Berechnung der Verfügbarkeit zusammengesetzter Systeme

Beschreibungsmethoden

- Daten- und Kontrollflussdiagramme
- Struktogramme
- Petrinetze
 - Netzwerkanalyse

Realzeitnachweis (RZN)

- RZN bei zeitgesteuerten Systemen
- RZN bei prioritätengesteuertem Scheduling
- RZN bei Deadline-Scheduling

Lernziele

Zeitliche Charakterisierung von Systemen

- Außenwelt (technische Prozesse, Anforderungen),
- Tasksets (Rechenprozesse, Lösungen) und
- ausgewähltes Rechensystems.

Moderne Realzeitsysteme entwerfen und realisieren können.

- Basistechnologien kennen (beispielsweise Multicore- versus Multikernel).
- Formale Beschreibungsmethoden kennen.

Lernziele

Probleme beim (programmtechnischen) Umgang mit Zeiten kennen und lösen können.

- Ortsabhängigkeit
- Darstellungsformen
- Berechnungen mit Zeiten

Nachweisen, dass zeitliche Anforderungen unter allen Umständen eingehalten werden (Realzeitnachweis**).**

Vielen Dank.

Und viel Erfolg!