

Überblick

Rekapitulation

Rückblick
Ausblick

Einleitung

Echtzeitbetrieb eines Rechensystems in seiner Umgebung

- ▶ schwache, starke oder strikte Echtzeitbedingungen

Fallbeispiel (Wärmetauscher)

- ▶ Schritt- und Antwortfunktion, Abtastrate, Zeitparameter
- ▶ Schwankungen in den Messergebnissen, Totzeit

Anforderungen funktionaler und nicht-funktionaler Art

- ▶ Aufzeichnung, Vergütung, Überwachung (dig. Kontrolle), Interaktion
- ▶ Rechtzeitigkeit und Verlässlichkeit

Klassifikation nach äußeren und inneren Faktoren

- ▶ *hard/soft real-time, fail safe/operational*
- ▶ *guaranteed response/best effort, resource adequate/inadequate*
- ▶ *event/time triggered*

eingebettete Systeme sind spezielle Echtzeitsysteme

Grundlagen

Einplanungseinheit \mapsto Prozedur, Faden und/oder Fadengruppe

- ▶ Aufgaben (*Tasks*) von Arbeitsaufträgen (*Jobs*)
- ▶ Verwaltungsgemeinkosten ein- und mehrfädiger Ausgaben
- ▶ verdrängbare und/oder nicht-verdrängbare Prozessinstanzen

Programmunterbrechung in synchroner oder asynchroner Ausprägung

- ▶ Zustandssicherung, Verwaltungsgemeinkosten des schlimmsten Falls
- ▶ **Interrupts** machen determinierte Programme nicht-deterministisch
 - ▶ nicht zu jedem Zeitpunkt ist bestimmt, wie weitergefahren wird
- ▶ Unvorhersagbarkeit, Überlast, Verzögerung, . . . , Nebenläufigkeit

Unterbrechungstechnik \mapsto Pegelsteuerung & Flankensteuerung

- ▶ Problem der Wiederbehauptung flankengesteuerter *Interrupts*
- ▶ maskierbare und nicht-maskierbare Unterbrechungen
- ▶ kaskadierbare bzw. kaskadierte Unterbrechungen

Ablaufsteuerung

Trennung unterschiedlicher Belange \mapsto Strategie & Mechanismus

- ▶ Einplanung ist die Strategie, Einlastung ist der Mechanismus

Arbeitsweise ist zeit- oder ereignisgesteuert: Einplanung & Einlastung

- ▶ entkoppelt im zeitgesteuerten System (Taktsteuerung)
- ▶ gekoppelt im ereignisgesteuerten System (Vorrangsteuerung)

Zeitparameter sind Punkte und Intervalle auf der Echtzeitachse

- ▶ (sporadische) Auslösezeit, (absoluter) Termin
- ▶ Antwortzeit bzw. relativer Termin, Schlupfzeit

Taskmodelle für periodische Aufgaben

- ▶ aperiodische oder sporadische Aufgaben bzw. Arbeitsaufträge
 - ▶ je nach dem, ob Jobtermine weich/fest oder hart sind
- ▶ Rangfolgen, Abhängigkeiten, Koordinierung, Verdrängung

Verfahren EDF, LRT, LST, RM und DM

Taktsteuerung

Ablauf Tabellen \mapsto vorberechnete (statische) Ablaufpläne

- ▶ Tabelleneinträge sind Jobs und deren Einlastungszeitpunkte

Einlastung und Laufzeitkontrolle im Abfrage- oder Unterbrecherbetrieb

- ▶ Taktzähler, Zeitgeber, Zeitkontrolle; Job als Routine/Koroutine

Struktur zyklischer Ablaufpläne \leadsto „gute Anordnung“, Determinismus

- ▶ Rahmen, Rahmenlänge, Scheiben; *major/minor cycle*

nichtperiodische Arbeitsaufträge \mapsto periodische/sporadische Jobs

- ▶ Schlupf (stehlen); Einplanung, Antwortzeitverbesserung, Einlastung
- ▶ Übernahmepfung (engl. *acceptance test*), Ausnahmebehandlung

Betriebswechsel bewerkstelligen aperiodische oder sporadische Jobs

- ▶ Tabellenwechsel, Betriebsmittelfreigabe/-anforderung, Nachladen

Vorrangsteuerung

Prioritäten gebunden an Ereignisauslöser oder -verarbeiter

- ▶ fest auf Task- und/oder Jobebene, ggf. dynamisch auf Taskebene
- ▶ Einplanungsaufwand: Auslöse- vs. Auswahlzeitpunkt von Jobs

Prioritätsabbildung mangels Systemprioritäten bzw. Prioritätsebenen

- ▶ gleichmäßig oder ungleichmäßig, *constant ratio mapping*
- ▶ Einfluss auf die Planbarkeit (engl. *schedulability*) eines Systems

Repräsentation von Ablaufplänen \mapsto Prioritätsschlange auf Tabellenbasis

- ▶ indizierte Bereitliste (*multi-level queue*), Aufwand begrenzen
- ▶ $O(1)$ -Scheduler: Jobsauslösung vs. Jobauswahl

Prioritätsverletzung und Prioritätsumkehr Selbstaussetzung/Hemmung

- ▶ kritische Abschnitte, unteilbare Betriebsmittel, Fernaufrufe
- ▶ Abstraktion „*considered harmful*“, nichtfunktionale Eigenschaften

Zugriffskontrolle

Konkurrenz und Koordination nebenläufiger Aktivitäten

- ▶ Nebenläufigkeit, Kausalität, Kausalordnung
- ▶ Konfliktsituationen \leadsto **synchronisieren ohne Prioritätsumkehr**

Verdrängungssteuerung \mapsto verdrängungsfreie kritische Abschnitte

- ▶ benötigt kein *à priori* Wissen; Verklemmungsvorbeugung
- ▶ pragmatisch/effektiv, beeinträchtigt unabhängige Jobs

Prioritätsvererbung \mapsto Priorität zeitweise erhöhen

- ▶ benötigt kein *à priori* Wissen
- ▶ direkte Blockierung, Blockierung durch Vererbung; transitiv

Prioritätsobergrenzen \mapsto Priorität zeitweise deckeln

- ▶ benötigt *à priori* Wissen; Verklemmungsvorbeugung
- ▶ Grundmodell vs. (einfachere) stapelorientierte Variante

Zustellerkonzepte

Problemfälle \mapsto Hintergrund-/Unterbrecherbetrieb, Schlupf

- ▶ periodische Verarbeitung nichtperiodischer Arbeitsaufträge
- ▶ Korrektheit und Optimalität von Ablaufplänen

periodische Zusteller (engl. *periodic server*)

- ▶ periodische Überprüfung der Warteschlange aperiodischer Jobs
- ▶ Ausführungsbudget, Verbrauchs- und Auffüllregeln

Bandweite verlierende Zusteller \mapsto Ausführungsbudget

- ▶ abfragender Zusteller (engl. *polling server, poller*)
- ▶ Verfall des Restbudgets zum Zeitpunkt des Untätigkeitswerdens

Bandweite erhaltende Zusteller (engl. *bandwidth-preserving server*)

- ▶ aufschiebbarer Zusteller (engl. *deferrable server*)
- ▶ einfacher sporadischer Zusteller (engl. *simple sporadic server*)

Architektur

Erscheinungsform \mapsto **verteiltes Echtzeitrechensystem**

- ▶ Netzwerkschnittstelle, Kommunikationssteuerung
- ▶ externe vs. autonome Kontrolle; Ereignis- vs. Zustandsnachrichten

Zusammensetzbarkeit (engl. *composability*)

- ▶ interagierende Gerätegruppen, Kompositionsproblem
- ▶ ereignis- vs. zeitgesteuerte Kommunikationssysteme

Skalierbarkeit (engl. *scalability*)

- ▶ Erweiterbarkeit (aber auch Schrumpfung), Komplexität
- ▶ Abkapselung von Zeitverhalten, Eigenständigkeit

Verlässlichkeit (engl. *dependability*)

- ▶ Fehlereingrenzung, Replikation, Replikdeterminismus
- ▶ Sicherheitshüllen, fehlereindämmende Regionen, Zertifizierung

Kommunikation

Anforderungen an Echtzeitkommunikation

- ▶ Protokolllatenz, Fehlererkennung; physikalische Struktur
- ▶ Unterstützung für Zusammensetzbarkeit, Flexibilität

Flusskontrolle zur Vermeidung von Überlast

- ▶ explizit, implizit; Gegenüberstellung (HRTS)
- ▶ *send and wait* (PAR), Aktionsverzögerung, *Thrashing*

Netzwerkarchitekturen „*OSI considered harmful*“?

- ▶ ISO OSI 7-Schichtenmodell, Architektur von Echtzeitnetzwerken
- ▶ Echtzeitnetzwerk, Feldbus, Rückgratnetz

Netzzugangsprotokolle zur Vergabe des Betriebsmittels „Bus“

- ▶ Kommunikationsmedium, Charakteristiken des Übertragungskanal
- ▶ CSMA (CD/CA), *Token*, *Minislotting*, *Master/Slave*, TDMA

Echtzeitsysteme (EZS) 2

Integrierte Lehrveranstaltung, 4 SWS

Inhalt

- ▶ Vorstellung von Systemplattform und Entwicklungsumgebung
 - ▶ Prozessoren/ μ -Controller: Tricore, ARM, AVR, MSP430
 - ▶ Betriebssysteme: eCos, ProOSEK, ProOSEKtime, QNX, TinyOS
- ▶ Analyse, Entwurf und Implementierung eines Echtzeitsystems
 - ▶ Bestimmung von WCETs, Prioritätenvergabe, Ablauf Tabellen, Treiber

Experimente

- ▶ Eisenkugel („Haut den Lukas“): Beschleunigen, Bremsen, Pendeln
- ▶ Modelleisenbahnanlage: Stellwerk, Zugführung
- ▶ Modellauto (LKW): Fernsteuerung; μ -Controller, CAN
- ? Kfz-Funktion (Audi): Fensterheber, Schiebedach, Blinker, ...

Teams von max. drei Personen, ggf. mehrere Teams pro Experiment

Fehlertoleranz in verteilten (Echtzeit-) Systemen

Hauptseminar, 2 SWS

26.04. Fehlertypen

03.05. Replikationstechniken

10.05. *N-Version Programming*

17.05. Einigungsalgorithmen

24.05. *Virtual Synchrony Model*

31.05. *Checkpointing*

14.06. Fehlerdetektoren

21.06. Injektion von Fehlern

28.06. Fehlertoleranz in Betriebssystemen

05.07. Fehlertoleranz in Delta 4

12.07. Fehlertoleranz in zeitgesteuerter Echtzeitarchitektur

19.07. Fehlertoleranz in *Peer-to-Peer*-Systemen

Studien-, Diplom- . . . Doktorarbeiten

Forschungs- und Entwicklungsprojekte: Universität, Forschungseinrichtungen, Industrie

