# Echtzeitsysteme Einleitung

18. Oktober 2010

© wosch

WS 2010/11

Echtzeitsysteme

2 Einleitung 2.1 Echtzeitbetrieb

#### DIN 44300

Ereignis- oder zeitgesteuerte Programmverarbeitung

- ▶ Echtzeitbetrieb ist ein Betrieb eines Rechensystems, bei dem Programme zur Verarbeitung anfallender Daten ständig betriebsbereit sind derart, dass die Verarbeitungsergebnisse innerhalb einer vorgegebenen Zeitspanne verfügbar sind.
- ▶ Die Daten können je nach Anwendungsfall nach einer zeitlich zufälligen Verteilung oder zu vorbestimmten Zeitpunkten anfallen.

2 Einleitung

### Überblick

#### Einleitung

**Echtzeitbetrieb** Fallbeispiel Wärmetauscher Zusammenfassung Bibliographie

WS 2010/11 Echtzeitsysteme 2-1 © wosch

2 Einleitung 2.1 Echtzeitbetrieb

# Verarbeitung von Programmen in Echtzeit

Realzeitverarbeitung (engl. real-time processing)

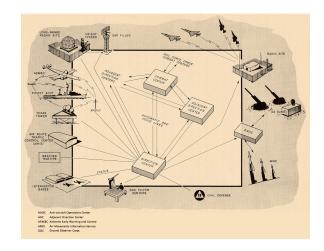
Zustandsänderung von Programmen wird zur Funktion der realen Zeit [?]

- ▶ korrektes Verhalten des Systems hängt nicht nur von den logischen Ergebnissen von Berechnungen ab
- ▶ zusätzlicher Aspekt ist der physikalische Zeitpunkt der Erzeugung und Verwendung der Berechnungsergebnisse
- Whirlwind (MIT, 1951), AN/FSQ-7 (Whirlwind II, IBM, 1957)
- SAGE (semi-automatic ground environment, 1958–1983)

© wosch WS 2010/11 Echtzeitsysteme 2-2 © wosch WS 2010/11 Echtzeitsysteme 2-3

# SAGE und AN/FSQ-7 "Whirlwind II"

Erstes Echtzeitrechensystem als Schöpfung des "Kalten Krieges"



- 27 Installationen über die USA verteilt
  - Nonstop-Betrieb
  - ▶ 25 Jahre
- durch Datenfernleitungen miteinander gekoppelt
  - ▶ Telefonleitungen
  - ► Internet-.. Mutter"
- pro Installation...
  - ▶ 100 Konsolen
  - ▶ 500 KLOC Ass.

2-4 © wosch WS 2010/11 Echtzeitsvsteme

2 Einleitung 2.1 Echtzeitbetrieb

# Echtzeitfähigkeit bedeutet Rechtzeitigkeit

Zuverlässige Reaktion des Rechensystems auf Umgebungsereignisse

Geschwindigkeit liefert keine Garantie, um rechtzeitig Ergebnisse von Berechnungen abliefern und Reaktionen darauf auslösen zu können

- ▶ asynchrone Programmunterbrechungen (engl. interrupts) können unvorhersagbare Laufzeitvarianzen verursachen
- > schnelle Programmausführung ist bestenfalls hinreichend für die rechtzeitige Bearbeitung einer Aufgabe

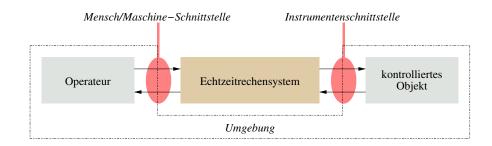
#### Zeit ist keine intrinsische Eigenschaft des Rechensystems

- ▶ die im Rechensystem verwendete Zeitskala muss mit der durch die Umgebung vorgegebenen identisch sein
- die Terminvorgaben der Umgebung gelten als weich, fest oder hart

2 Einleitung 2.1 Echtzeitbetrieb

### Komponenten eines Echtzeitsystems

Echtzeitrechensystem und seine Umgebung



- ▶ das Echtzeitrechensystem muss auf Stimuli innerhalb von Zeitspannen reagieren, die durch die Umgebung vorgegeben sind
- ▶ der Zeitpunkt, zu dem ein Ergebnis vorliegen muss, wird als Termin oder Frist (engl. deadline) bezeichnet

© wosch WS 2010/11 2-5 Echtzeitsvsteme

2 Einleitung 2.1 Echtzeitbetrieb

## Arten von Echtzeitsystemen

Verbindlichkeit von Terminvorgaben

weich (engl. soft) auch "schwach"

- ▶ das Ergebnis einer zu einem vorgegebenen Termin nicht geleisteten Arbeit ist weiterhin von Nutzen
- Terminverletzung ist tolerierbar

fest (engl. firm) auch "stark"

- ▶ das Ergebnis einer zu einem vorgegebenen Termin nicht geleisteten Arbeit ist wertlos und wird verworfen
- ▶ Terminverletzung ist tolerierbar, führt zum Arbeitsabbruch

hart (engl. hard) auch "strikt"

- ▶ das Versäumnis eines fest vorgegebenen Termins kann eine "Katastrophe" hervorrufen
- ▶ Terminverletzung ist keinesfalls tolerierbar

© wosch WS 2010/11 Echtzeitsvsteme 2-6 © wosch WS 2010/11 Echtzeitsvsteme 2-7 2 Einleitung 2.1 Echtzeitbetrieb

# Arten von Echtzeitsystemen (Forts.)

 $\mathsf{Fest} \Longleftrightarrow \mathsf{Hart}$ 

 $\mathsf{fest}/\mathsf{hart} \, \mapsto \, \mathsf{Terminverletzung} \, \, \mathsf{ist} \, \, \mathsf{nicht} \, \, \mathsf{ausgeschlossen}^1$ 

▶ die Terminverletzung wird vom Betriebssystem erkannt

 $\mathsf{fest} \sim \mathsf{plangem\"{a}B}$  weiterarbeiten

- das Betriebssystem bricht den Arbeitsauftrag ab
- der nächste Arbeitsauftrag wird gestartet
- ► ist transparent für die Anwendung

hart → sicheren Zustand finden

- das Betriebssystem löst eine Ausnahmesituation aus
- die Ausnahmebehandlung führt zum sicheren Zustand
- ► ist intransparent für die Anwendung

© wosch WS 2010/11 Echtzeitsysteme 2-8

2 Einleitung 2.1 Echtzeitbetrieb

## Vorhersagbarkeit des Laufzeitverhaltens

Echtzeitsysteme sind (schwach, stark oder strikt) deterministisch

#### Determiniertheit

Bei ein und derselben Eingabe sind verschiedene Abläufe zulässig, alle Abläufe liefern jedoch stets das gleiche Resultat.

- ► Transparenz von Programmunterbrechungen
  - Interrupts verursachen vom "normalen Ablauf" verschiedene ausnahmebedingte Abläufe
  - ► Verzögerungen in der Programmausführung wirken sich nicht nachteilig aus
- ► Terminvorgaben der Umgebung werden eingehalten

#### Determinismus

Zu jedem Zeitpunkt ist bestimmt, wie weitergefahren wird.

2 Einleitung 2.1 Echtzeitbetrieb

### Arten von Echtzeitsystemen (Forts.)

Radikale Unterschiede im Systementwurf zeichnen sich ab...

#### hard real-time computer system

- ein Rechensystem, das mind. einen strikten Termin erreichen muss
  - ▶ garantiert unter allen (spezifizierten) Last- und Fehlerbedingungen
  - das Laufzeitverhalten ist ausnahmslos deterministisch
- ▶ typisch für ein sicherheitskritisches Echtzeitrechensystem
  - engl. safety-critical real-time computer system

#### soft real-time computer system

- ein Rechensystem, das keinen strikten Termin erreichen muss
- es ist erlaubt, gelegentlich Termine zu verpassen

© wosch WS 2010/11 Echtzeitsysteme 2-9

2 Einleitung 2.1 Echtzeitbetrieb

# Spezialzweckbetrieb

Verhalten von Echtzeitanwendungen [?, S. 25]

rein zyklisch  $\sim$  nur periodische Tasks, Polling-Betrieb

- ▶ nahezu konstanter Betriebsmittelbedarf von Periode zu Periode meist zyklisch → überwiegend periodische *Tasks*
- ightharpoonup das System muss auf externe Ereignisse reagieren können asynchron und irgendwie vorhersagbar  $\sim$  kaum periodische *Tasks* 
  - ▶ aufeinanderfolgende Aktivierungen können zeitlich stark variieren
  - ▶ Zeitdifferenzen haben eine obere Grenze oder bekannte Statistik

asynchron und nicht vorhersagbar → aperiodische Tasks

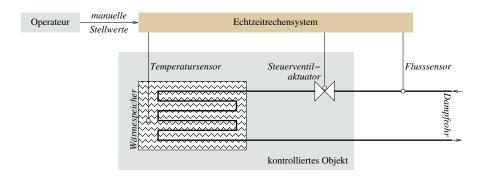
- ► Anwendungen reagieren auf asynchrone Ereignisse
- ▶ hohe, nicht deterministische Laufzeitkomplexität einzelner *Tasks*

© wosch WS 2010/11 Echtzeitsysteme 2 - 10 © wosch WS 2010/11 Echtzeitsysteme 2 - 11

<sup>&</sup>lt;sup>1</sup>Auch wenn Ablaufplan und Betriebssystem auf dem Blatt Papier Determinismus zeigen, kann das im Feld eingesetzte technische System von Störeinflüssen betroffen sein, die ggf. die Verletzung auch eines harten Termins nach sich ziehen.

### Aufbau des Demonstrators

Elementare Kontrollschleife



Aufgabe des Echtzeitrechensystems: Kontrolle des Steuerventils

- ▶ Dampffluss durch den Wärmetauscher bestimmen
- ▶ Temperatur im Wärmespeicher im vorgewählten Bereich halten

2-12 © wosch Echtzeitsvsteme

2 Einleitung 2.2 Fallbeispiel Wärmetauscher

# Kontrollierendes Rechensystem

Echtzeitrechensystem

Temperaturwerte sind periodisch abzutasten, um Abweichungen des aktuellen Wertes von dem eingestellten Wert zu erkennen:

d<sup>sample</sup> Zeitabstand (konstant) zwischen zwei Abtastungen

- ▶ analoge auf digitale Werte abbilden ~> A/D-Wandlung
  - diskretes System sich quasi-kontinuierlich verhalten lassen
- ▶ Faustregel:  $d^{sample} < (d^{rise}/10)$

 $f^{sample}$  Abtastfrequenz, entspricht  $1/d^{sample}$ 

Abweichung (Ist-/Sollwert) bestimmen und dem Regelungsalgorithmus zur Berechnung des neuen Stellwertes zuführen:

d<sup>cpu</sup> Zeitdauer bis zur Ausgabe des neuen Stellwertes

- ▶ digitale auf analoge Werte abbilden ~> D/A-Wandlung
- ▶ Randbedingung:  $d^{cpu} < d^{sample}$

 $\Delta d^{cpu}$  Differenz zwischen Minimum und Maximum von  $d^{cpu}$ 

Kontrolliertes Objekt

Schrittfunktion (engl. step function) und Antwortfunktion (engl. response function)

Erhöhung des Dampfflusses (Schrittfunktion) verändert die Temperatur im Wärmespeicher (Antwortfunktion), bis Gleichgewicht eingestellt ist:

▶ die Objektdynamik bestimmt sich durch die Flüssigkeitsmenge im Wärmespeicher und den Dampffluss des Wärmetauschers

Zeitparameter zur Charakterisierung der Schritt-/Antwortfunktion:

dobject Zeitdauer bis die Temperatur zu steigen beginnt

- ▶ hervorgerufen durch die (initiale) Trägheit des Objektes
- ▶ auch als Prozessverzögerung (engl. process lag) bezeichnet

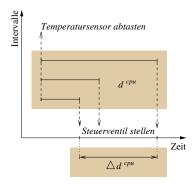
drise Zeitdauer bis zum (erneuten) Temperaturgleichgewicht

2-13 © wosch WS 2010/11 Echtzeitsvsteme

2 Einleitung 2.2 Fallbeispiel Wärmetauscher

# Kontrollierendes Rechensystem (Forts.)

Schwankung (engl. jitter) in den Messergebnissen



ist trotz konstantem Rechenaufwand zur Stellwertbestimmung variabel

- verdrängende Einplanung
- ▶ überlappende Ein-/Ausgabe
- ► Programmunterbrechungen
- Busüberlastung, DMA

∆d<sup>cpu</sup> fügt Unschärfe zum Zeitpunkt der Temperatursensorabtastung hinzu

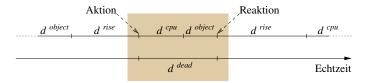
- bewirkt zusätzlichen Fehler
- ▶ beeinträchtigt die Dienstgüte
- unbekannte variable Verzögerungen können bei der Regelung nicht kompensiert werden, aber bekannte konstante Verzögerungen
- ▶ Randbedingung:  $\Delta d^{cpu} \ll d^{cpu}$

© wosch WS 2010/11 Echtzeitsysteme 2-14 © wosch WS 2010/11 Echtzeitsvsteme 2-15 2 Einleitung 2.2 Fallbeispiel Wärmetauscher

#### 

d<sup>dead</sup> Zeitintervall zwischen Start der Aktion zur Stellwertberechnung und Wahrnehmung einer Reaktion nach erfolgter Steuerung

- $\triangleright$  setzt sich zusammen aus  $d^{cpu}$  und  $d^{object}$ , d.h.:
  - 1. der Implementierung des kontrollierenden Rechensystems
  - 2. der Dynamik des kontrollierten Objektes



- ▶ beeinträchtigt Güte und Stabilität der Kontrollschleife
  - ▶ insbesondere in Anbetracht der mit *d<sup>cpu</sup>* gegebenen Varianz
- ▶ gibt einem relative Ungewissheit über die erzielte Wirkung

© wosch WS 2010/11 Echtzeitsysteme 2 - 16

2 Einleitung 2.4 Bibliographie

#### Literaturverzeichnis

© wosch WS 2010/11 Echtzeitsysteme 2 - 18

#### 2 Einleitung 2.3 Zusammenfassung

### Resümee

#### Echtzeitbetrieb eines Rechensystems in seiner Umgebung

- ► Komponenten eines Echtzeitsystems:
  - ▶ Operateur, Echtzeitrechensystem, kontrolliertes Objekt
- ▶ Verhalten von Echtzeitanwendungen:
  - ► rein/meist zyklisch
  - asynchron und irgendwie/nicht vorhersagbar
- ▶ schwache, starke oder strikte Echtzeitbedingungen

#### Fallbeispiel (Wärmetauscher)

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

© wosch WS 2010/11 Echtzeitsysteme 2-17