

# Echtzeitsysteme

## Einleitung

Peter Ulbrich

Lehrstuhl Informatik 4

06. Oktober 2014

# Gliederung

## 1 Historischer Bezug

- Das erste Echtzeitrechensystem
- SAGE – Der Nachfolger
- Heutige Echtzeitsysteme

## 2 Echtzeitbetrieb

- Definition
- Realzeitbetrieb
- Termine
- Deterministische Ausführung

## 3 Aufbau und Abgrenzung

- Struktur dieser Vorlesung
- Abgrenzung

## 4 Zusammenfassung

# Whirlwind I

## Das erste Echtzeitrechensystem

**Zweck:** Flugsimulator  
(Ausbildung von Bomberbesatzungen)

**Auftraggeber:** U.S. Navy

**Auftragnehmer:** MIT

**Laufzeit:** 1945 – 1952



(Quelle: Alex Handy from Oakland, Nmibia)

**Bauweise:** Digitalrechner, bit-parallele Operationen, 5000 Röhren, 11000 Halbleiterdioden, magnetischer Kernspeicher, Röhrenmonitore, Lichtgriffel

**später:** Nutzung durch die U.S. Air Force im SAGE



# SAGE und AN/FSQ-7 alias „Whirlwind II“

Daten AN/FSQ-7 alias „Whirlwind II“:



SAGE Bedienstation

Auftraggeber: U.S. Air Force

Auftragnehmer: MIT, später IBM

Bauweise: 55000 Röhren, 2000  $m^2$ , 275  
Tonnen, 3 MW, 75 KIPS

(Quelle: Steve Jurvetson from Menlo Park, USA)

Betriebsdaten SAGE:

Installation: 22 - 23 Stationen im Zeitraum 1959 - 1963

Betrieb: bis 1983 (Whirlwind I bis 1979)

Nachfolgesysteme: z.B. AWACS

Kosten: 8 – 12 Milliarden \$ (1964), entspricht ca. 55 Milliarden \$ (2000)

# Moderne Echtzeitsysteme

Wo immer Rechensysteme mit ihrer physikalischen Umwelt interagieren ...



# Spezialzweckssysteme (Forts.)

## Verteiltes System auf Rädern

### CAN CLASS B

- ① SAM-GRS Fahrer
- ② SAM-GRS Beifahrer
- ③ SAM-GRS Heck 1
- ④ SAM-GRS Heck 2
- ⑤ Sitzsteuergerät Fahrer
- ⑥ Sitzsteuergerät Beifahrer
- ⑦ Sitzsteuergerät hinten links
- ⑧ Sitzsteuergerät hinten rechts
- ⑨ Türsteuergerät vorne Fahrerseite
- ⑩ Türsteuergerät vorne Beifahrerseite
- ⑪ Türsteuergerät hinten Fahrerseite
- ⑫ Türsteuergerät hinten Beifahrerseite
- ⑬ Steuerung Thermovent
- ⑭ Dachbedieneinheit
- ⑮ Dachknoten Mitte (DKM)
- ⑯ Vorderes-Boden-Feld (VBF)
- ⑰ Hinteres-Boden-Feld (HBF)
- ⑱ Elektronisches Zündschloss (EZS)
- ⑲ Kombiinstrument
- ⑳ Motorleuchtmodul
- ㉑ Frontklimatisierung
- ㉒ Fondklimatisierung
- ㉓ Audiogateway

- ㉔ Parktronicssystem (PTS)
- ㉕ Rollendruckkontrolle (RDK)
- ㉖ Pneumatische Steuerung (PSE)
- ㉗ Heckdifferenzialschleissung-öffnung
- ㉘ Zentrales Gateway
- ㉙ Airbag-EG (Armada)
- ㉚ Multifunktionssteuergerät (MSS)
- ㉛ Bordnetz-Steuergerät
- ㉜ Wandler Lenkschleifung
- ㉝ Standheizung
- ㉞ Türanzheizung hinten Fahrerseite
- ㉟ Türanzheizung hinten Beifahrerseite

### CAN CLASS C

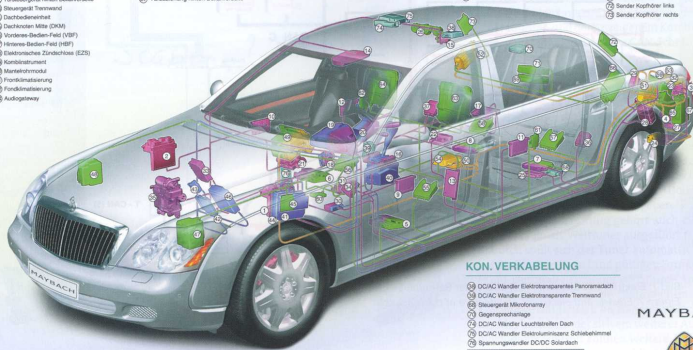
- ⑲ Elektronisches Zündschloss (EZS)
- ⑳ Kombiinstrument
- ㉑ Motorleuchtmodul
- ㉒ Zentrales Gateway
- ㉓ Elektronisches Wählhebelmodul
- ㉔ Luftfederung (SLF)
- ㉕ Detronic (DTR)
- ㉖ Leuchtwagenregulierung
- ㉗ Motorelektronik (ME)
- ㉘ Servotronic Brake System (FSG)
- ㉙ Elektronische Getriebe-Steuerung

### MOST-BUS

- ㉚ Audiogateway
- ㉛ Headunit
- ㉜ Steuergerät Sprachbedienung
- ㉝ TV-Tuner MOST
- ㉞ Soundverstärker
- ㉟ Navigationsrechner
- ㊱ Kommunikationsplattform (CPI)

### PRIVATE-BUS

- ① Sitzsteuergerät Fahrer
- ② Sitzsteuergerät Beifahrer
- ③ Sitzsteuergerät hinten links
- ④ Sitzsteuergerät hinten rechts
- ⑤ TV-Tuner CAN
- ⑥ Dachinstrument
- ⑦ Servotronic Brake System (FSG)
- ⑧ Servotronic Brake System (ASG 2)
- ⑨ Multikonturlehne vorne links
- ⑩ Multikonturlehne vorne rechts
- ⑪ Multikonturlehne hinten links
- ⑫ Multikonturlehne hinten rechts
- ⑬ Keyless Go Heckmodul
- ⑭ Keyless Go Innenraummodul
- ⑮ Keyless Go Tür hinten links
- ⑯ Keyless Go Tür hinten rechts
- ⑰ Fondklimatisierung
- ⑱ Fondklimatisierung rechts
- ⑲ Kommunikationsplattform Ford (CPI)
- ⑳ Surround Amplifier
- ㉑ Audio Video Controller
- ㉒ CO-Wachzeile
- ㉓ DVD-Spieler
- ㉔ Sender Kopfhörer links
- ㉕ Sender Kopfhörer rechts



### KON. VERKABELUNG

- ㉚ DC/AC Wandler Elektrotransparentes Panoramadach
- ㉛ DC/AC Wandler Elektrotransparente Thermovent
- ㉜ Steuergerät Motorleuchtmodul
- ㉝ Gegenschleifung
- ㉞ DC/AC Wandler Leuchtdiagnostik
- ㉟ DC/AC Wandler Elektrotransparenz Schiebehimmel
- ㊱ Spannungswandler DC/DC Solarlatch

Σ aller Steuergeräte: 76

MAYBACH



(Quelle: DaimlerChrysler [1])

# Gliederung

- 1 Historischer Bezug
  - Das erste Echtzeitrechensystem
  - SAGE – Der Nachfolger
  - Heutige Echtzeitsysteme
- 2 Echtzeitbetrieb
  - Definition
  - Realzeitbetrieb
  - Termine
  - Deterministische Ausführung
- 3 Aufbau und Abgrenzung
  - Struktur dieser Vorlesung
  - Abgrenzung
- 4 Zusammenfassung



# 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.*

# 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.*

# Komponenten eines Echtzeitsystems

## Echtzeitrechnungssystem und seine Umgebung



Operateur



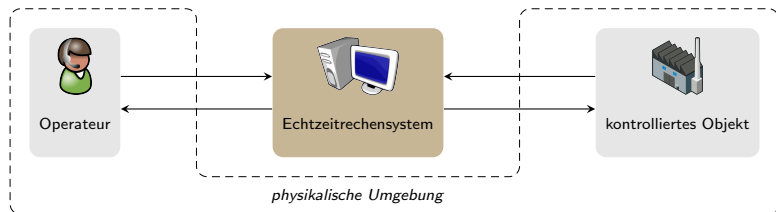
Echtzeitrechnungssystem



kontrolliertes Objekt

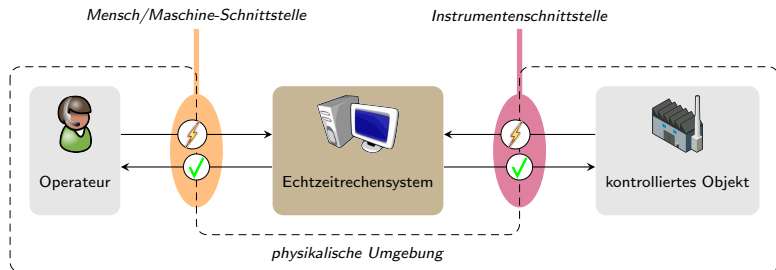
# Komponenten eines Echtzeitsystems

## Echtzeitrechnungssystem und seine Umgebung



# Komponenten eines Echtzeitsystems

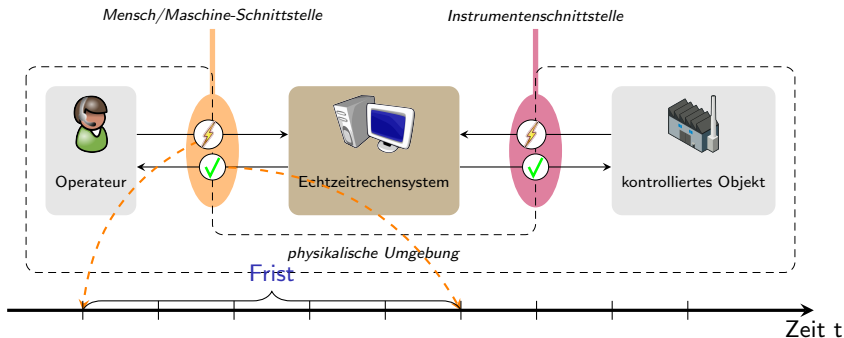
## Echtzeitrechnungssystem und seine Umgebung



- das Echtzeitrechnungssystem berechnet als Reaktion auf Stimuli bzw. **Ereignisse** (engl. *event*) der Umgebung **Ergebnisse**

# Komponenten eines Echtzeitsystems

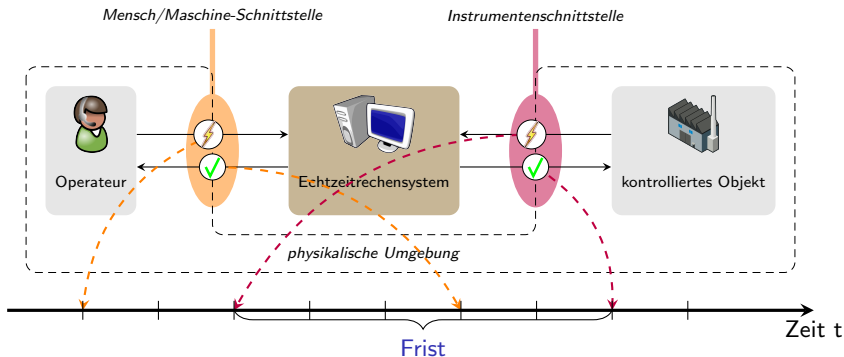
## Echtzeitrechnungssystem und seine Umgebung



- das Echtzeitrechnungssystem berechnet als Reaktion auf Stimuli bzw. **Ereignisse** (engl. *event*) der Umgebung **Ergebnisse**
- der Zeitpunkt, zu dem ein Ergebnis vorliegen muss, wird als **Termin** oder **Frist** (engl. *deadline*) bezeichnet

# Komponenten eines Echtzeitsystems

## Echtzeitrechnungssystem und seine Umgebung



- das Echtzeitrechnungssystem berechnet als Reaktion auf Stimuli bzw. **Ereignisse** (engl. *event*) der Umgebung **Ergebnisse**
- der Zeitpunkt, zu dem ein Ergebnis vorliegen muss, wird als **Termin** oder **Frist** (engl. *deadline*) bezeichnet

# Verarbeitung von Programmen in Echtzeit

Realzeitverarbeitung (engl. *real-time processing*)

Echtzeitbetrieb bedeutet **Rechtzeitigkeit**

- korrektes Systemverhalten hängt nicht nur von den logischen Ergebnissen der Berechnungen ab
- zusätzlicher Aspekt ist der **physikalische Zeitpunkt** der Erzeugung und Verwendung der Berechnungsergebnisse
- den Rahmen stecken der Eintrittspunkt des Ereignisses und die entsprechende Frist ab



# Verarbeitung von Programmen in Echtzeit

Realzeitverarbeitung (engl. *real-time processing*)

Echtzeitbetrieb bedeutet **Rechtzeitigkeit**

- korrektes Systemverhalten hängt nicht nur von den logischen Ergebnissen der Berechnungen ab
- zusätzlicher Aspekt ist der **physikalische Zeitpunkt** der Erzeugung und Verwendung der Berechnungsergebnisse
- den Rahmen stecken der Eintrittspunkt des Ereignisses und die entsprechende Frist ab

☞ Termine hängen dabei von der Anwendung ab

**wenige Mikrosekunden** z.B. Drehzahl- und Stromregelung bei der Ansteuerung von Elektromotoren

**einige Millisekunden** z.B. Multimedia-Anwendungen  
(Übertragung von Ton- und Bildmaterial)

**Sekunden, Minuten, Stunden** z.B. Prozessanlagen (Erhitzen von Wasser)

# Geschwindigkeit impliziert nicht unbedingt 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

# Geschwindigkeit impliziert nicht unbedingt 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 Rechner system verwendete Zeitskala muss nicht mit der durch die Umgebung vorgegebenen identisch sein
- die zeitlichen Gegebenheiten des kontrollierten Objekts müssen im Rechner system geeignet abgebildet werden

# Konsequenzen überschrittener Termine

## Verbindlichkeit von Terminvorgaben

weich (engl. *soft*) auch „schwach“

- das Ergebnis verliert mit zunehmender Terminüberschreitung an Wert (z.B. Bildrate bei Multimediasystemen)
- Terminverletzung ist tolerierbar

fest (engl. *firm*) auch „stark“

- das Ergebnis wird durch eine Terminüberschreitung wertlos und wird verworfen (z.B. Abgabetermin einer Übungsaufgabe)
- Terminverletzung ist tolerierbar, führt zum Arbeitsabbruch

hart (engl. *hard*) auch „strikt“

- eine Terminüberschreitung kann zum Systemversagen führen und eine „Katastrophe“ hervorrufen (z.B. Airbag)
- Terminverletzung ist keinesfalls tolerierbar

# Konsequenzen überschrittener Termine

## Verbindlichkeit von Terminvorgaben

weich (engl. *soft*) auch „schwach“

- das Ergebnis verliert mit zunehmender Terminüberschreitung an Wert (z.B. Bildrate bei Multimediasystemen)
- Terminverletzung ist tolerierbar

fest (engl. *firm*) auch „stark“

- das Ergebnis wird durch eine Terminüberschreitung wertlos und wird verworfen (z.B. Abgabetermin einer Übungsaufgabe)
- Terminverletzung ist tolerierbar, führt zum Arbeitsabbruch

hart (engl. *hard*) auch „strikt“

- eine Terminüberschreitung kann zum Systemversagen führen und eine „Katastrophe“ hervorrufen (z.B. Airbag)
- Terminverletzung ist keinesfalls tolerierbar

# Konsequenzen überschrittener Termine

## Verbindlichkeit von Terminvorgaben

**weich** (engl. *soft*) auch „schwach“

- das Ergebnis verliert mit zunehmender Terminüberschreitung an Wert (z.B. Bildrate bei Multimediasystemen)
- Terminverletzung ist tolerierbar

**fest** (engl. *firm*) auch „stark“

- das Ergebnis wird durch eine Terminüberschreitung wertlos und wird verworfen (z.B. Abgabetermin einer Übungsaufgabe)
- Terminverletzung ist tolerierbar, führt zum Arbeitsabbruch

**hart** (engl. *hard*) auch „strikt“

- eine Terminüberschreitung kann zum Systemversagen führen und eine „Katastrophe“ hervorrufen (z.B. Airbag)
- Terminverletzung ist keinesfalls tolerierbar

# Konsequenzen überschrittener Termine

## Verbindlichkeit von Terminvorgaben

**weich** (engl. *soft*) auch „schwach“

- das Ergebnis verliert mit zunehmender Terminüberschreitung an Wert (z.B. Bildrate bei Multimediasystemen)
- Terminverletzung ist tolerierbar

**fest** (engl. *firm*) auch „stark“

- das Ergebnis wird durch eine Terminüberschreitung wertlos und wird verworfen (z.B. Abgabetermin einer Übungsaufgabe)
- Terminverletzung ist tolerierbar, führt zum Arbeitsabbruch

**hart** (engl. *hard*) auch „strikt“

- eine Terminüberschreitung kann zum Systemversagen führen und eine „Katastrophe“ hervorrufen (z.B. Airbag)
- Terminverletzung ist keinesfalls tolerierbar

# Arten von Echtzeitsystemen

Fest  $\longleftrightarrow$  Hart

fest/hart  $\mapsto$  Terminverletzung ist nicht ausgeschlossen<sup>1</sup>

- die Terminverletzung wird vom Betriebssystem erkannt

---

<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.



# Arten von Echtzeitsystemen

Fest  $\longleftrightarrow$  Hart

fest/hart  $\mapsto$  Terminverletzung ist nicht ausgeschlossen<sup>1</sup>

- die Terminverletzung wird vom Betriebssystem erkannt

fest  $\leadsto$  plangemäß weiterarbeiten

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

---

<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.

# Arten von Echtzeitsystemen

Fest  $\longleftrightarrow$  Hart

**fest/hart**  $\mapsto$  Terminverletzung ist nicht ausgeschlossen<sup>1</sup>

- die Terminverletzung wird vom Betriebssystem erkannt

**fest**  $\leadsto$  plangemäß weiterarbeiten

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

**hart**  $\leadsto$  sicheren Zustand finden

- das Betriebssystem löst eine **Ausnahmesituation** aus
- die Ausnahme ist **intransparent für die Anwendung**
- die **Anwendung** behandelt diese Ausnahme

---

<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.

# 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*

# Arten von Echtzeitsystemen (Forts.)

Radikale Unterschiede im Systementwurf zeichnen sich ab...

## *hard real-time computer system*

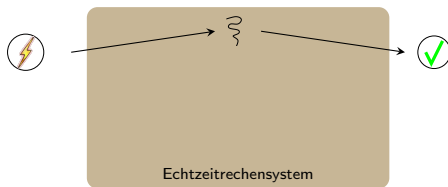
- ein Rechensystem, das mind. einen strikten Termin erreichen muss
  - garantiert unter allen (spezifizierten) Last- und Fehlerbedingungen
  - das Laufzeitverhalten ist ausnahmslos deterministisch
- typisch für ein **sicherheitskritisches Echtzeitrechensystem**
  - engl. *safety-critical real-time computer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

## *soft real-time computer system*

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

# Herausforderung: Gewährleisten von Rechtzeitigkeit

Ereignisbehandlungen müssen termingerecht abgearbeitet werden

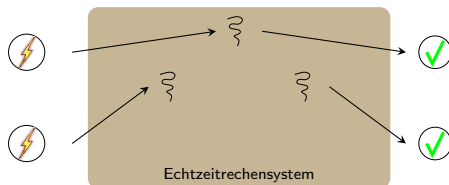


## Ereignisse aktivieren **Ereignisbehandlungen**

- Wie viel Zeit benötigt die Ereignisbehandlung **maximal**?
- Lösung des trivialen Falls ist (scheinbar) einfach, wenn man die **maximale Ausführungszeit** der Ereignisbehandlung kennt.

# Herausforderung: Gewährleisten von Rechtzeitigkeit

Ereignisbehandlungen müssen termingerecht abgearbeitet werden



## Ereignisse aktivieren Ereignisbehandlungen

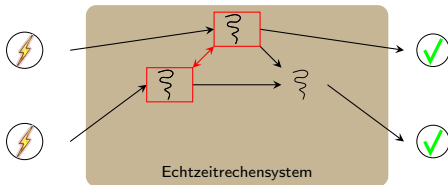
- Wie viel Zeit benötigt die Ereignisbehandlung **maximal**?
- Lösung des trivialen Falls ist (scheinbar) einfach, wenn man die **maximale Ausführungszeit** der Ereignisbehandlung kennt.

## Reale Echtzeitsysteme sind **komplex**

- mehrere Ereignisbehandlungen  $\leadsto$  Konkurrenz
  - Verwaltung gemeinsamer Betriebsmittel, allen voran die CPU.

# Herausforderung: Gewährleisten von Rechtzeitigkeit

Ereignisbehandlungen müssen termingerecht abgearbeitet werden



Ereignisse aktivieren **Ereignisbehandlungen**

- Wie viel Zeit benötigt die Ereignisbehandlung **maximal**?
- Lösung des trivialen Falls ist (scheinbar) einfach, wenn man die **maximale Ausführungszeit** der Ereignisbehandlung kennt.

Reale Echtzeitsysteme sind **komplex**

- mehrere Ereignisbehandlungen  $\leadsto$  Konkurrenz
  - Verwaltung gemeinsamer Betriebsmittel, allen voran die CPU.
- Abhängigkeiten zwischen verschiedenen Ereignisbehandlungen

# 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.*



# 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**

👉 **unzureichend**, falls zeitliche Aspekte bedeutend sind

# 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**

 **unzureichend**, falls zeitliche Aspekte bedeutend sind

## Determinismus

*Identische Eingaben führen zu identischen Abläufen. Zu jedem Zeitpunkt ist bestimmt, wie weitergefahren wird.*

# 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**

☞ **unzureichend**, falls zeitliche Aspekte bedeutend sind

## Determinismus

*Identische Eingaben führen zu identischen Abläufen. Zu jedem Zeitpunkt ist bestimmt, wie weitergefahren wird.*

- ☞ **notwendig**, falls Termine einzuhalten sind
- nur so lässt sich das Laufzeitverhalten verlässlich abschätzen

# Vorhersagbarkeit des Laufzeitverhaltens (Forts.)

Echtzeitsysteme sind (schwach, stark oder strikt) deterministisch

## Vorhersagbarkeit

*Der Ablauf lässt sich zu jedem Zeitpunkt exakt angeben und hängt nicht von den aktuellen Eingaben oder vom aktuellen Zustand ab.*

# Vorhersagbarkeit des Laufzeitverhaltens (Forts.)

Echtzeitsysteme sind (schwach, stark oder strikt) deterministisch

## Vorhersagbarkeit

*Der Ablauf lässt sich zu jedem Zeitpunkt exakt angeben und hängt nicht von den aktuellen Eingaben oder vom aktuellen Zustand ab.*

- von Umgebung und Eingaben entkoppeltes Laufzeitverhalten
  - Aktivitäten folgen einem strikt vorgegebenem Stundenplan
- 👉 **vorteilhaft** für zeitkritische Systeme
  - exakte Angaben zum zeitlichen Ablauf sind bereits à priori möglich

# Vorhersagbarkeit des Laufzeitverhaltens (Forts.)

Echtzeitsysteme sind (schwach, stark oder strikt) deterministisch

## Vorhersagbarkeit

*Der Ablauf lässt sich zu jedem Zeitpunkt exakt angeben und hängt nicht von den aktuellen Eingaben oder vom aktuellen Zustand ab.*

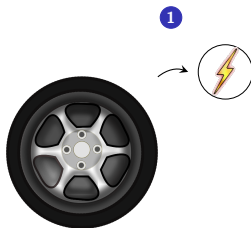
- von Umgebung und Eingaben entkoppeltes Laufzeitverhalten
  - Aktivitäten folgen einem strikt vorgegebenem Stundenplan
- 👉 **vorteilhaft** für zeitkritische Systeme
  - exakte Angaben zum zeitlichen Ablauf sind bereits à priori möglich

👉 Das Echtzeitsystem muss stets ein **deterministisches** oder besser **vorhersagbares** Laufzeitverhalten gewährleisten.

- insbesondere beim **Zugriff auf gemeinsame Betriebsmittel**  
**CPU**  $\mapsto$  Umschaltung zwischen verschiedenen Aktivitäten  
**Kommunikationsmedium**  $\mapsto$  Versand von Nachrichten

# Beispiel: ein (fiktives) AB-System

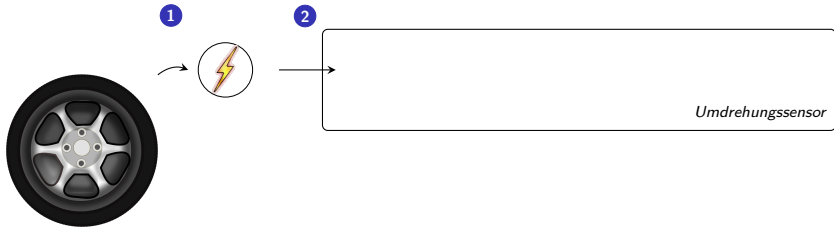
Ein Gemeinschaftserzeugnis eines verteilten Echtzeitrechensystems



- das ABS überwacht kontinuierlich Umdrehungszahl des Rads
  - so kann z. B. ein Blockieren des Rades erkannt werden

# Beispiel: ein (fiktives) AB-System

Ein Gemeinschaftserzeugnis eines verteilten Echtzeitrechensystems

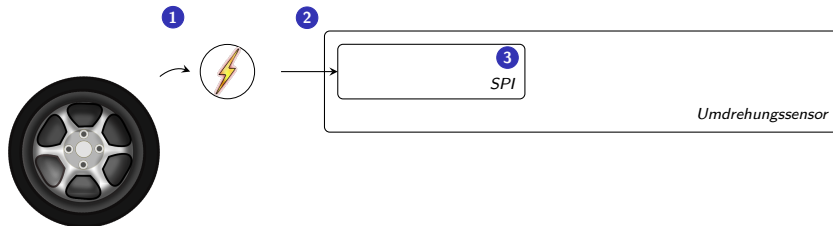


- das ABS überwacht kontinuierlich Umdrehungszahl des Rads
  - so kann z. B. ein Blockieren des Rades erkannt werden
- in einem **intelligenten Sensor** (engl. *smart sensor*) findet zunächst eine Vorverarbeitung der erfassten Daten statt



# Beispiel: ein (fiktives) AB-System

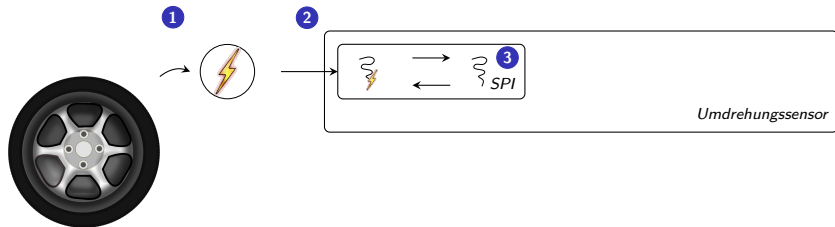
Ein Gemeinschaftserzeugnis eines verteilten Echtzeitrechensystems



- die Daten selbst werden über den SPI-Bus entgegengenommen

# Beispiel: ein (fiktives) AB-System

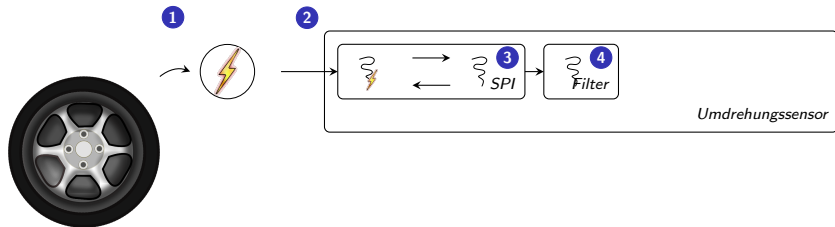
Ein Gemeinschaftserzeugnis eines verteilten Echtzeitrechensystems



- die Daten selbst werden über den SPI-Bus entgegengenommen
  - die Buskommunikation erfordert einen ISR und einen Faden
    - ~ Wann wird die ISR angesprungen? Sind Unterbrechungen gesperrt?
    - ~ Wann wird der Faden eingeplant? Muss er auf Betriebsmittel warten?

# Beispiel: ein (fiktives) AB-System

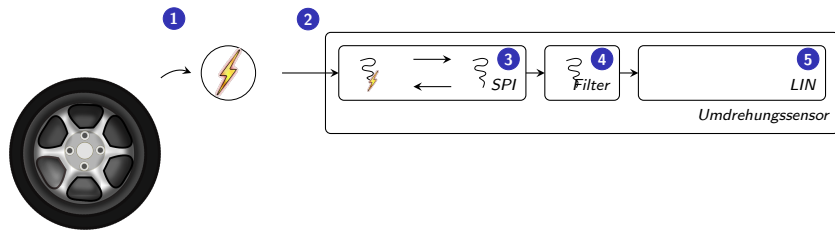
Ein Gemeinschaftserzeugnis eines verteilten Echtzeitrechnungssystems



- anschließend übernimmt ein Filter die Vorverarbeitung
    - Angleichung diverser Ausführungsraten durch den gesonderten Faden
      - der Filter verarbeitet immer mehrere Messwerte auf einmal
- ~ Wann wird der Faden eingeplant? Muss er auf Betriebsmittel warten?

# Beispiel: ein (fiktives) AB-System

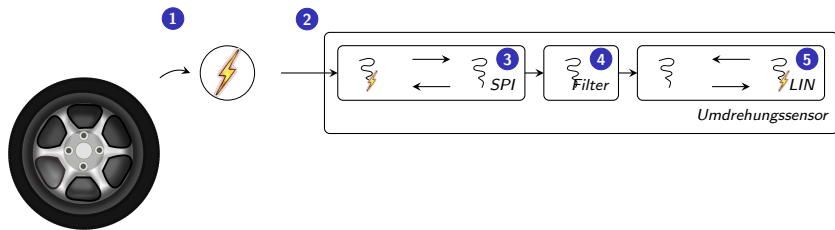
Ein Gemeinschaftserzeugnis eines verteilten Echtzeitrechensystems



- die Messwerte werden dann an das ABS-Steuergerät gesendet

# Beispiel: ein (fiktives) AB-System

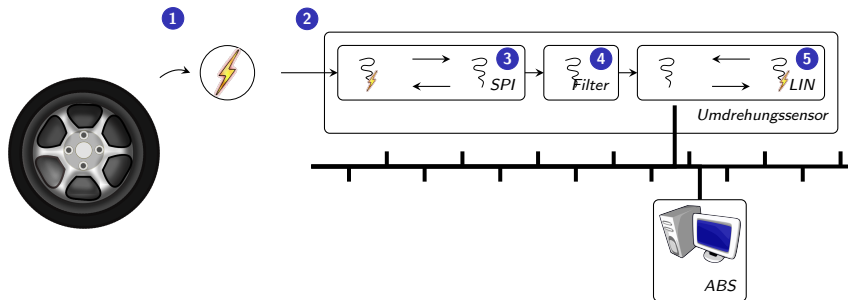
Ein Gemeinschaftserzeugnis eines verteilten Echtzeitrechensystems



- die Messwerte werden dann an das ABS-Steuergerät gesendet
  - auch hier ist ein komplexer Gerätetreiber notwendig
    - ~ Wann wird die ISR angesprungen? Sind Unterbrechungen gesperrt?
    - ~ Wann wird der Faden eingeplant? Muss er auf Betriebsmittel warten?
    - ~ Können alle Daten „auf einmal“ übertragen werden?

# Beispiel: ein (fiktives) AB-System

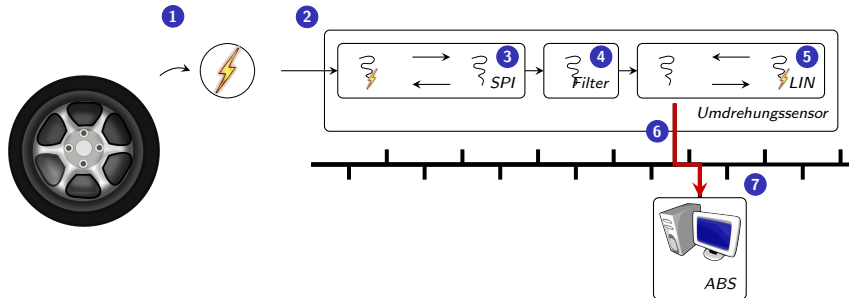
Ein Gemeinschaftserzeugnis eines verteilten Echtzeitrechnungssystems



- Sensor und ABS-Steuergerät sind über einen LIN-Bus verbunden

# Beispiel: ein (fiktives) AB-System

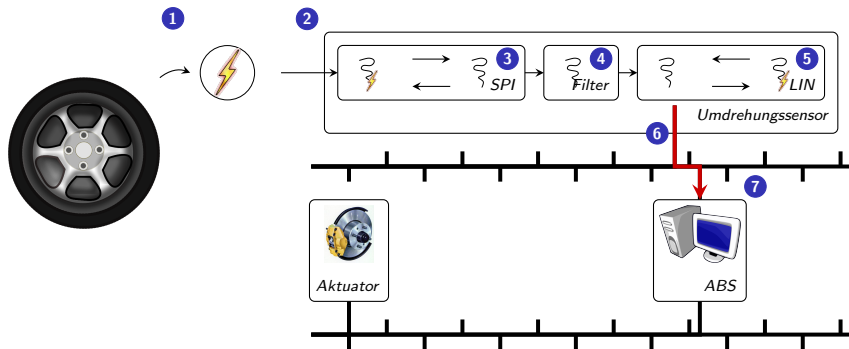
Ein Gemeinschaftserzeugnis eines verteilten Echtzeitrechnungssystems



- Sensor und ABS-Steuergerät sind über einen LIN-Bus verbunden
  - auch die Datenübertragung benötigt Zeit ...
    - ~> Wieviele Bytes schafft der Bus in einer bestimmten Zeit?
    - ~> Wie lange muss ich warten, bis ich auf das Medium zugreifen kann?
- die Vorgänge auf dem ABS-Steuergerät sind noch deutlich komplexer

# Beispiel: ein (fiktives) AB-System

Ein Gemeinschaftserzeugnis eines verteilten Echtzeitrechnungssystems

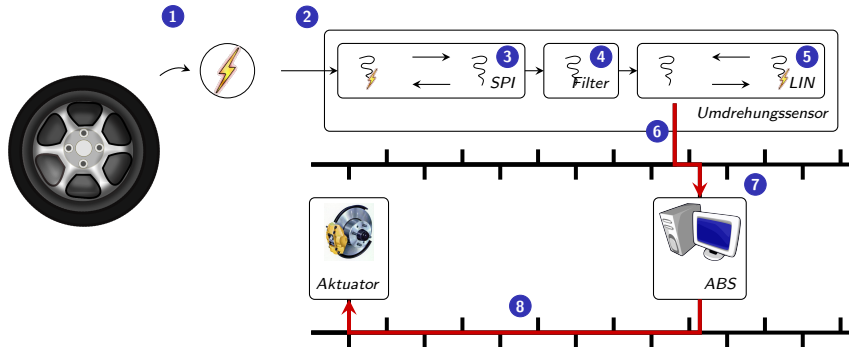


- der berechnete Stellwert wird dem Aktor zugestellt
  - dieser ist z. B. über einen CAN-Bus an das Steuergerät angebunden



# Beispiel: ein (fiktives) AB-System

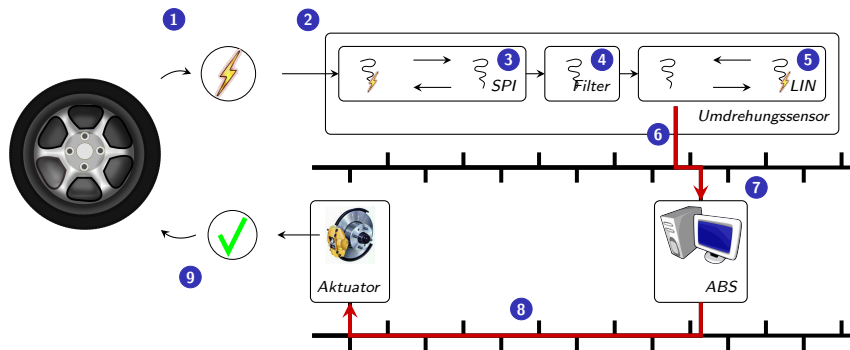
Ein Gemeinschaftserzeugnis eines verteilten Echtzeitrechensystems



- der berechnete Stellwert wird dem Aktor zugestellt
  - dieser ist z. B. über einen CAN-Bus an das Steuergerät angebunden
    - ~> Wieviele Bytes schafft der Bus in einer bestimmten Zeit?
    - ~> Wie lange muss ich warten, bis auf das Medium zugreifen kann?

# Beispiel: ein (fiktives) AB-System

Ein Gemeinschaftserzeugnis eines verteilten Echtzeitrechensystems



- der berechnete Stellwert wird dem Aktor zugestellt
  - dieser ist z. B. über einen CAN-Bus an das Steuergerät angebunden
    - ~ Wieviele Bytes schafft der Bus in einer bestimmten Zeit?
    - ~ Wie lange muss ich warten, bis auf das Medium zugreifen kann?
- ☞ schließlich wird die Bremskraft geeignet beeinflusst

# Beispiel: ein (fiktives) AB-System (Forts.)

Wie lange dauert das ganze nun?

Für eine korrekte Funktion des AB-Systems muss die Reaktion auf eine Blockierung des Rades **in einer bestimmten Zeitspanne** gewährleistet sein.

# Beispiel: ein (fiktives) AB-System (Forts.)

Wie lange dauert das ganze nun?

Für eine korrekte Funktion des AB-Systems muss die Reaktion auf eine Blockierung des Rades **in einer bestimmten Zeitspanne** gewährleistet sein. Zu dieser Zeitspanne tragen zwei Komponenten bei:

„aktive“ Zeitintervalle  $\leadsto$  „Fortschritt“ im ABS

- Berechnungen benötigen Zeit  $\leadsto$  **maximale Ausführungszeit**
- Geschwindigkeit der Datenübertragung ist beschränkt

# Beispiel: ein (fiktives) AB-System (Forts.)

Wie lange dauert das ganze nun?

Für eine korrekte Funktion des AB-Systems muss die Reaktion auf eine Blockierung des Rades **in einer bestimmten Zeitspanne** gewährleistet sein. Zu dieser Zeitspanne tragen zwei Komponenten bei:

„aktive“ Zeitintervalle  $\leadsto$  „Fortschritt“ im ABS

- Berechnungen benötigen Zeit  $\leadsto$  **maximale Ausführungszeit**
- Geschwindigkeit der Datenübertragung ist beschränkt

„inaktive“ Zeitintervalle  $\leadsto$  „Wartezeit“ für das ABS

- Fortschritt erfordert die Zuteilung von Betriebsmitteln
  - z. B. der CPU für eine Berechnung oder des Kommunikationsmediums für die Übertragung der Daten

# Beispiel: ein (fiktives) AB-System (Forts.)

Wie lange dauert das ganze nun?

Für eine korrekte Funktion des AB-Systems muss die Reaktion auf eine Blockierung des Rades **in einer bestimmten Zeitspanne** gewährleistet sein. Zu dieser Zeitspanne tragen zwei Komponenten bei:

„aktive“ Zeitintervalle  $\leadsto$  „Fortschritt“ im ABS

- Berechnungen benötigen Zeit  $\leadsto$  **maximale Ausführungszeit**
- Geschwindigkeit der Datenübertragung ist beschränkt

„inaktive“ Zeitintervalle  $\leadsto$  „Wartezeit“ für das ABS

- Fortschritt erfordert die Zuteilung von Betriebsmitteln
  - z. B. der CPU für eine Berechnung oder des Kommunikationsmediums für die Übertragung der Daten

- ☞ Die Frage ist, wie lange man auf die Zuteilung warten muss!
- **Determiniertheit** alleine reicht für die Beantwortung nicht aus!
  - **Determinismus** erfordert die vollständige Kenntnis der Umgebung!
  - **Vorhersagbarkeit** liefert die gewünschte Aussage zu dieser Frage!

# Spezialzweckbetrieb

Verhalten von Echtzeitanwendungen [4, S. 25]

## deterministische Abarbeitung von Ereignisbehandlungen

rein zyklisch  $\leadsto$  nur periodische Ereignisbehandlungen, Abfrage-Betrieb

- nahezu konstanter Betriebsmittelbedarf von Periode zu Periode

meist zyklisch  $\leadsto$  überwiegend periodische Ereignisbehandlungen

- das System muss auf externe Ereignisse reagieren können

asynchron/vorhersagbar  $\leadsto$  kaum periodische Ereignisbehandlungen


- aufeinanderfolgende Aktivierungen können zeitlich stark variieren
- Zeitdifferenzen haben eine obere Grenze oder bekannte Statistik

asynchron/nicht vorhersagbar  $\leadsto$  aperiodische Ereignisbehandlungen

- Anwendungen reagieren auf asynchrone Ereignisse
- hohe, nicht deterministische Laufzeitkomplexität einzelner Ereignisbehandlungen

# Spezialzweckbetrieb

Verhalten von Echtzeitanwendungen [4, S. 25]

 deterministische Abarbeitung von Ereignisbehandlungen

**rein zyklisch**  $\leadsto$  nur periodische Ereignisbehandlungen, Abfrage-Betrieb

- nahezu konstanter Betriebsmittelbedarf von Periode zu Periode

**meist zyklisch**  $\leadsto$  überwiegend periodische Ereignisbehandlungen

- das System muss auf externe Ereignisse reagieren können

**asynchron/vorhersagbar**  $\leadsto$  kaum periodische Ereignisbehandlungen

- aufeinanderfolgende Aktivierungen können zeitlich stark variieren
- Zeitdifferenzen haben eine obere Grenze oder bekannte Statistik


**asynchron/nicht vorhersagbar**  $\leadsto$  aperiodische Ereignisbehandlungen

- Anwendungen reagieren auf asynchrone Ereignisse
- hohe, nicht deterministische Laufzeitkomplexität einzelner Ereignisbehandlungen



# Spezialzweckbetrieb

Verhalten von Echtzeitanwendungen [4, S. 25]

 deterministische Abarbeitung von Ereignisbehandlungen

**rein zyklisch**  $\leadsto$  nur periodische Ereignisbehandlungen, Abfrage-Betrieb

- nahezu konstanter Betriebsmittelbedarf von Periode zu Periode

**meist zyklisch**  $\leadsto$  überwiegend periodische Ereignisbehandlungen

- das System muss auf externe Ereignisse reagieren können

**asynchron/vorhersagbar**  $\leadsto$  kaum periodische Ereignisbehandlungen


- aufeinanderfolgende Aktivierungen können zeitlich stark variieren
- Zeitdifferenzen haben eine obere Grenze oder bekannte Statistik

**asynchron/nicht vorhersagbar**  $\leadsto$  aperiodische Ereignisbehandlungen

- Anwendungen reagieren auf asynchrone Ereignisse
- hohe, nicht deterministische Laufzeitkomplexität einzelner Ereignisbehandlungen

# Spezialzweckbetrieb

Verhalten von Echtzeitanwendungen [4, S. 25]

 deterministische Abarbeitung von Ereignisbehandlungen

**rein zyklisch**  $\leadsto$  nur periodische Ereignisbehandlungen, Abfrage-Betrieb

- nahezu konstanter Betriebsmittelbedarf von Periode zu Periode

**meist zyklisch**  $\leadsto$  überwiegend periodische Ereignisbehandlungen

- das System muss auf externe Ereignisse reagieren können

**asynchron/vorhersagbar**  $\leadsto$  kaum periodische Ereignisbehandlungen


- aufeinanderfolgende Aktivierungen können zeitlich stark variieren
- Zeitdifferenzen haben eine obere Grenze oder bekannte Statistik

**asynchron/nicht vorhersagbar**  $\leadsto$  aperiodische Ereignisbehandlungen

- Anwendungen reagieren auf asynchrone Ereignisse
- hohe, nicht deterministische Laufzeitkomplexität einzelner Ereignisbehandlungen

# Spezialzweckbetrieb

Verhalten von Echtzeitanwendungen [4, S. 25]

 deterministische Abarbeitung von Ereignisbehandlungen

**rein zyklisch**  $\leadsto$  nur periodische Ereignisbehandlungen, Abfrage-Betrieb

- nahezu konstanter Betriebsmittelbedarf von Periode zu Periode

**meist zyklisch**  $\leadsto$  überwiegend periodische Ereignisbehandlungen

- das System muss auf externe Ereignisse reagieren können

**asynchron/vorhersagbar**  $\leadsto$  kaum periodische Ereignisbehandlungen

- aufeinanderfolgende Aktivierungen können zeitlich stark variieren
- Zeitdifferenzen haben eine obere Grenze oder bekannte Statistik

**asynchron/nicht vorhersagbar**  $\leadsto$  aperiodische Ereignisbehandlungen

- Anwendungen reagieren auf asynchrone Ereignisse
- hohe, nicht deterministische Laufzeitkomplexität einzelner Ereignisbehandlungen

# Gliederung

- 1 Historischer Bezug
  - Das erste Echtzeitrechensystem
  - SAGE – Der Nachfolger
  - Heutige Echtzeitsysteme
- 2 Echtzeitbetrieb
  - Definition
  - Realzeitbetrieb
  - Termine
  - Deterministische Ausführung
- 3 Aufbau und Abgrenzung
  - Struktur dieser Vorlesung
  - Abgrenzung
- 4 Zusammenfassung

# Aufbau der Vorlesung

Die Vorlesung orientiert sich vor allem ...

- an der Ausprägung des Spezialzweckbetriebs ...

Einleitung

Grundlagen

# Aufbau der Vorlesung

Die Vorlesung orientiert sich vor allem ...

- an der Ausprägung des Spezialzweckbetriebs ...

Einleitung

Grundlagen

vorranggesteuerte  
Systeme

taktgesteuerte  
Systeme

Analyse

# Aufbau der Vorlesung

Die Vorlesung orientiert sich vor allem ...

- an der Ausprägung des Spezialzweckbetriebs ...
- und den Eigenschaften der Ereignisse und ihrer Behandlungen,

Einleitung			
Grundlagen			
	vorranggesteuerte Systeme	taktgesteuerte Systeme	Analyse
periodische Echtzeitsysteme			
nicht-periodische Echtzeitsysteme			
Rangfolge			
Zugriffskontrolle			

# Aufbau der Vorlesung

Die Vorlesung orientiert sich vor allem ...

- an der Ausprägung des Spezialzweckbetriebs ...
- und den Eigenschaften der Ereignisse und ihrer Behandlungen,
- blickt aber auch über den Tellerrand.

Einleitung

Grundlagen

vorranggesteuerte  
Systeme

taktgesteuerte  
Systeme

Analyse

periodische Echtzeitsysteme

nicht-periodische Echtzeitsysteme

Mehrkernprozessor-Echtzeitsysteme\*

Rangfolge

Zugriffskontrolle

**Exkurs / Industrievortrag**



# Aufbau der Vorlesung

Die Vorlesung orientiert sich vor allem ...

- an der Ausprägung des Spezialzweckbetriebs ...
- und den Eigenschaften der Ereignisse und ihrer Behandlungen,
- blickt aber auch über den Tellerrand.

Einleitung

Grundlagen

vorranggesteuerte  
Systeme

taktgesteuerte  
Systeme

Analyse

periodische Echtzeitsysteme

nicht-periodische Echtzeitsysteme

Mehrkernprozessor-Echtzeitsysteme\*

Rangfolge

Zugriffskontrolle

**Exkurs / Industrievortrag**

Zusammenfassung und Ausblick

# Abgrenzung

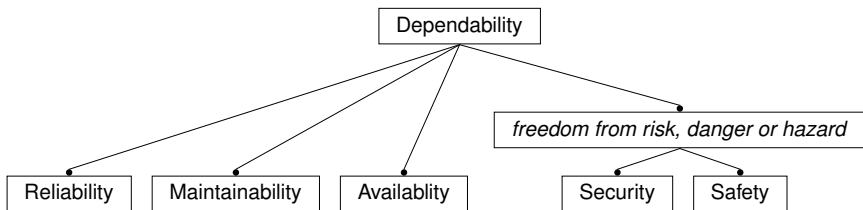
Verlässlichkeit (engl. *dependability*)

Echtzeitsysteme sind häufig **sicherheitskritische Systeme** und erfordern ein hohes Maß an **Verlässlichkeit**.

# Abgrenzung

Verlässlichkeit (engl. *dependability*)

Echtzeitsysteme sind häufig **sicherheitskritische Systeme** und erfordern ein hohes Maß an **Verlässlichkeit**. Verlässlichkeit selbst hat viele Gesichter ...



*The trustworthiness of a computing system which allows reliance to be justifiably placed on the service it delivers. [3]*

# Abgrenzung

## Zusammenspiel von Rechtzeitigkeit und Verlässlichkeit

Verlässlichkeit **erfordert** Rechtzeitigkeit!

- Verpasste Termine stellen Fehler dar.
- Diese Fehler müssen ggf. erkannt oder maskiert werden.

# Abgrenzung

## Zusammenspiel von Rechtzeitigkeit und Verlässlichkeit

Verlässlichkeit **erfordert** Rechtzeitigkeit!

- Verpasste Termine stellen Fehler dar.
- Diese Fehler müssen ggf. erkannt oder maskiert werden.

**Andererseits:** Rechtzeitigkeit **erfordert** Verlässlichkeit!

- Fehler können zum Verpassen eines Termins führen.
- Maskieren solcher Fehler hilft, die Rechtzeitigkeit zu gewährleisten.

# Abgrenzung

## Zusammenspiel von Rechtzeitigkeit und Verlässlichkeit

Verlässlichkeit **erfordert** Rechtzeitigkeit!

- Verpasste Termine stellen Fehler dar.
- Diese Fehler müssen ggf. erkannt oder maskiert werden.

**Andererseits:** Rechtzeitigkeit **erfordert** Verlässlichkeit!

- Fehler können zum Verpassen eines Termins führen.
- Maskieren solcher Fehler hilft, die Rechtzeitigkeit zu gewährleisten.

 Betrachtung der Rechtzeitigkeit unter Annahme des *fehlerfreien Falls*

- Verletzte Termine werden auf einer höheren Ebene behandelt.
- Toleranz gegenüber Fehlern dient der Verlässlichkeit.
  - Entsprechende Maßnahmen zum Erreichen von Fehlertoleranz werden also nicht durch harte Termine impliziert.

# Abgrenzung


## Zusammenspiel von Rechtzeitigkeit und Verlässlichkeit

Verlässlichkeit **erfordert** Rechtzeitigkeit!


- Verpasste Termine stellen Fehler dar.
- Diese Fehler müssen ggf. erkannt oder maskiert werden.

**Andererseits:** Rechtzeitigkeit **erfordert** Verlässlichkeit!

- Fehler können zum Verpassen eines Termins führen.
- Maskieren solcher Fehler hilft, die Rechtzeitigkeit zu gewährleisten.

 Betrachtung der Rechtzeitigkeit unter Annahme des *fehlerfreien Falls*

- Verletzte Termine werden auf einer höheren Ebene behandelt.
- Toleranz gegenüber Fehlern dient der Verlässlichkeit.
  - Entsprechende Maßnahmen zum Erreichen von Fehlertoleranz werden also nicht durch harte Termine impliziert.

 Harte Echtzeitsysteme sind häufig auch äußerst verlässlich.

;-)

# Gliederung

- 1 Historischer Bezug
  - Das erste Echtzeitrechensystem
  - SAGE – Der Nachfolger
  - Heutige Echtzeitsysteme
- 2 Echtzeitbetrieb
  - Definition
  - Realzeitbetrieb
  - Termine
  - Deterministische Ausführung
- 3 Aufbau und Abgrenzung
  - Struktur dieser Vorlesung
  - Abgrenzung
- 4 Zusammenfassung



# Resümee

Echtzeitbetrieb eines Rechensystems in seiner Umgebung

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

Abgrenzung Fokus dieser Vorlesung liegt auf der Rechtzeitigkeit

- Rechtzeitigkeit und Verlässlichkeit bedingen sich oft gegenseitig
- Maßnahmen zu ihrem Erreichen sind grundverschieden:
  - Verlässlichkeit  $\rightsquigarrow$  Robustheit durch Fehlertoleranz
  - Rechtzeitigkeit  $\rightsquigarrow$  deterministisches Ablaufverhalten

# Resümee

## Echtzeitbetrieb eines Rechensystems in seiner Umgebung

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

## Abgrenzung Fokus dieser Vorlesung liegt auf der Rechtzeitigkeit

- Rechtzeitigkeit und Verlässlichkeit bedingen sich oft gegenseitig
- Maßnahmen zu ihrem Erreichen sind grundverschieden:
  - Verlässlichkeit  $\leadsto$  Robustheit durch Fehlertoleranz
  - Rechtzeitigkeit  $\leadsto$  deterministisches Ablaufverhalten

# Resümee

**Echtzeitbetrieb** eines Rechensystems in seiner Umgebung

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

**Abgrenzung** Fokus dieser Vorlesung liegt auf der **Rechtzeitigkeit**

- Rechtzeitigkeit und Verlässlichkeit bedingen sich oft gegenseitig
- Maßnahmen zu ihrem Erreichen sind grundverschieden:
  - Verlässlichkeit  $\rightsquigarrow$  Robustheit durch Fehlertoleranz
  - Rechtzeitigkeit  $\rightsquigarrow$  deterministisches Ablaufverhalten

# Literaturverzeichnis

- [1] DAIMLERCHRYSLER AG:  
Der neue Maybach.  
In: *ATZ/MTZ Sonderheft* (2002), Sept., S. 125
- [2] DEUTSCHES INSTITUT FÜR NORMUNG:  
*DIN 44300: Informationsverarbeitung — Begriffe.*  
Berlin, Köln : Beuth-Verlag, 1985
- [3] IFIP:  
*Working Group 10.4 on Dependable Computing and Fault Tolerance.*  
<http://www.dependability.org/wg10.4>, 2003
- [4] LIU, J. W. S.:  
*Real-Time Systems.*  
Prentice-Hall, Inc., 2000. –  
ISBN 0-13-099651-3