Echtzeitbetriebssysteme

Oliver Jack

Ernst-Abbe-Hochschule Jena
Fachbereich Elektrotechnik und Informationstechnik

Sommersemester 2025

Ernst-Abbe-Hochschule Jena
University of Applied Sciences

Lerneinheit Einführung

Lerneinheit Einführung

Themen der Lehrveranstaltung

Grundlagen von Betriebssystemen und Echtzeitbetriebssystemen

- Einsatzgebiete von Echtzeitbetriebssystemen
- Systemarchitekturen,
- Prozesse, Tasks, Threads,
- Warteschlangenkonzepte (Scheduling),
- Signale und Ausnahmen
- Entwurfsmethodik für Echtzeitanwendungen

Qualifikationsziele

Sie sollen die wesentlichen Konzepte und Grundlagen von Echtzeitbetriebssystemen kennen und Fertigkeiten in der Anwendung und im Einsatz solcher Systeme erlangen.

- Kenntnisse von Konzepten der Ressourcenverwaltung und Ablaufsteuerung in Echtzeitsystemen
- Fähigkeiten und Fertigkeiten zur Durchführung der Anforderungsanalyse, zur Auswahl und zum Einsatz eines Echtzeitbetriebssystem für eine Modellanwendung
- Kenntnisse von Laufzeitmodellen
- Kenntnisse von Interprozesskommunikation und Warteschlangenkonzepten
- Sicherer Umgang mit der Terminologie im Bereich der Echtzeitbetriebssysteme

Organisation

Lehrformen

- Vorlesung
- Übung
- Online-Lernmaterial
- Abgabe von Aufgaben und Projektarbeit im Moodle-Kurs

Ggf. Termine

https://stundenplan.eah-jena.de/moses/index.html

Organisatorisches (Forts.)

Kommunikation

- Prof. Dr.-Ing. Oliver Jack
- Tel. 0 36 41 / 20 57 15
- E-Mail: Oliver.Jack@eah-jena.de
- Website: `https://www.eah-jena.de/et/personen/oliver-jack`
- Büro: 05.02.09
- Sprechzeiten: gemäß Website
- Aktuelle Informationen zur Veranstaltung im Moodle-Kurs

Organisatorisches (Forts.)

Übung

- Anwendung des Lehrstoffes
- Handwerkliche Übung
- Projektgruppenarbeit
- Semesterprojekt

Modulprüfung

- Alternative Prüfungsleistung: Semesterprojekt, Abgabe und Präsentation am letzten Praktikumstermin
- Wiederholungsmöglichkeit: im Turnus des Lehrveranstaltungsangebots
- Achtung: Note ist Bestandteil des Abschlusszeugnisses!

Literatur

D.L. Buhr, R.J.A.and Bailey.
An Introduction to Real-Time Systems: From Design to Multitasking with C/C++.
Prentice Hall, Upper Saddle River, 1998.

Giorgio Buttazzo, Giuseppe Lipari, Luca Abeni, and Marco Caccamo.
Soft Real-Time Systems: Predictability vs. Efficiency.
Springer, New York, 2005.

Hermann Kopetz.
Real-Time Systems. Design Principles for Distributed Embedded Applications.
Kluwer Academic Publishers, Dordrecht, London, 1997.

Literatur

Phillip A. Laplante.
Real-Time Systems Design and Analysis.
IEEE Computer Society Press, Los Alamitos, second edition, 1997.

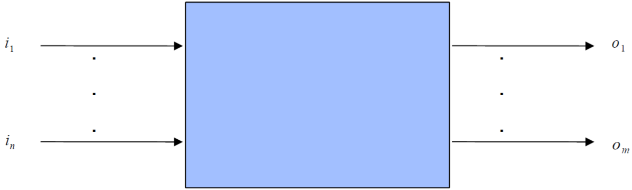
Jane W.S. Liu.
Real-Time Systems.
Prentice Hall, Upper Saddle River, 2. edition, 2000.

Dieter Zöbel and Wolfgang Albrecht.
Echtzeitsysteme: Grundlagen und Techniken.
International Thomson Publishing, Bonn, 1995.

Lerneinheit Einführung

System

- A system has a set of one or more inputs entering a black box and a set of one or more outputs exiting the black box.
- System with n inputs and m outputs



- If $i_1, \ldots, i_n \in I_1 \times \cdots \times I_n$ (input space) and $o_1, \ldots, o_m \in O_1 \times \cdots \times O_m$ (output space) then a System S is a subset of the overall cross product:

$$S \subseteq I_1 \times \cdots \times I_n \times O_1 \times \cdots \times O_m$$

Deterministic System

Definition

A system is said to be deterministic if for each possible state, and each set of inputs, a unique set of outputs, response times and next state of the system can be determined.

Event determinism

Next states and outputs of the system are known for each set of inputs which trigger events.

Temporal determinism

The response time of each set of outputs is known.

Real-Time

Real-Time System

> *Any system in which the time at which the output is produced is significant. This is usually because the input corresponds to some movement in the physical world, and the output has to relate to that same movement. The lag from input time to output time must be sufficiently small for acceptable timeliness*

[Oxford dictionary of Computing]

Real-Time System

Real-Time system is defined as a system where the correctness of the system depends not only the result of computations but also on the time at which it is produced. Therefore the time is the most important item to be managed.

Real-Time Classification

Hard Real-Time

Systems where failure to meet system response time constraints leads to a system failure are called hard real-time systems.

Soft Real-Time

Systems where performance is degraded but not destroyed by failure to meet system response time constraints.

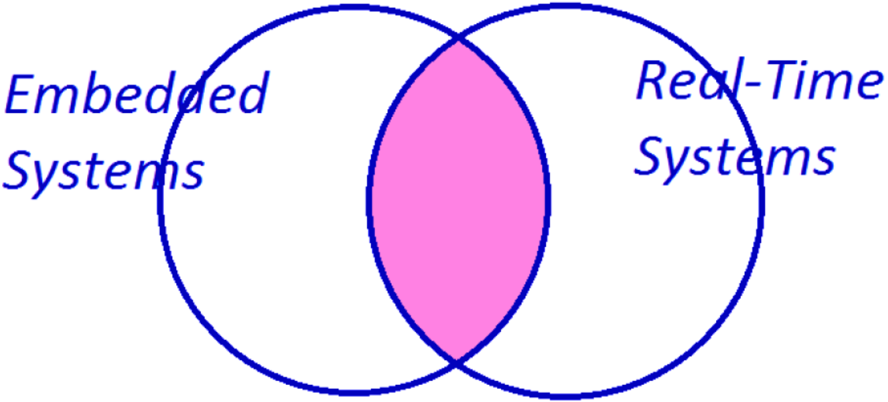
Firm Real-Time

Systems with hard deadlines where some low probability of missing deadline can be tolerated.

Examples of Hard, Soft and Firm RT Systems

System	Class	Explanation
Automated teller machine	soft	missing even many deadlines will not lead to catastrophic failure, only degraded performance.
Embedded navigation controller for autonomous weed killer	firm	missing critical navigation deadlines causes the robot to veer out of control and damage crops.
Emergency brake on a high speed train	hard	Missing even one deadline will delay stopping the train and could lead to a disaster.

Real-Time and Embedded Systems



Main Characteristics of Real-Time Systems

- Determinism in terms of time
- Reliability
- Dependability

Dependability

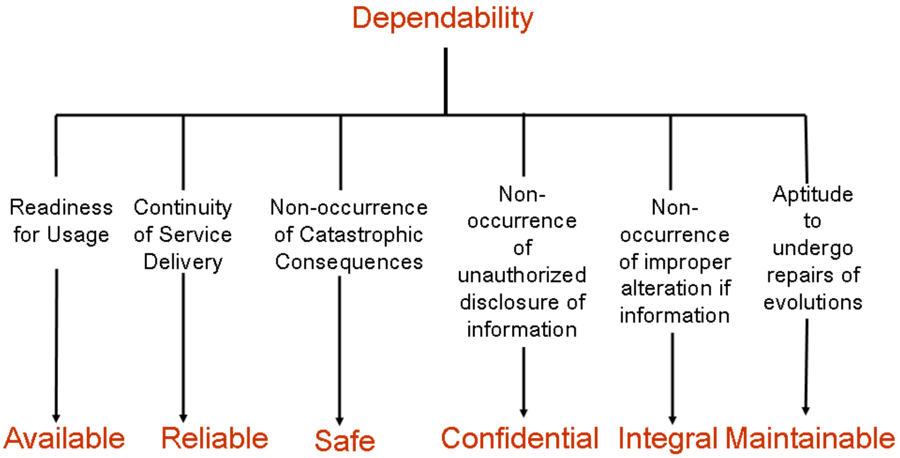
- Dependability: the property of a computing system which allows reliance to be placed on the service it delivers;
- System failure: occurs when the delivered service deviates from service stated by the specification;
- An error is that part of the system state which is liable to lead to failure;
- A fault is an adjudged cause of an error;
- An error is thus the manifestation of a fault in the system and a failure is the effect of an error on the service.

Common Terms

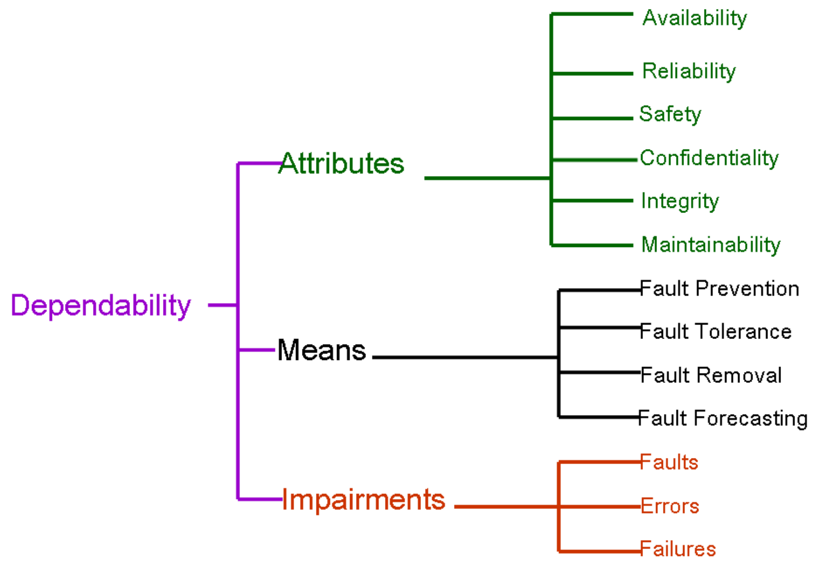
In order to achieve a dependable computing system, a number of diverse techniques can be used at various stages of the system design Probably the most successful method of achieving reliable systems is to use a combination of one or more of the following:

- Fault Avoidance: how to prevent, by construction, fault occurrence or introduction;
- Fault Tolerance: how to provide, by redundancy, a service complying with the specification in spite of faults;
- Fault Removal: how to minimize, by verification, the presence of faults;
- Fault Forecasting: how to estimate, by evaluation, the presence, the creation and the consequences of faults.

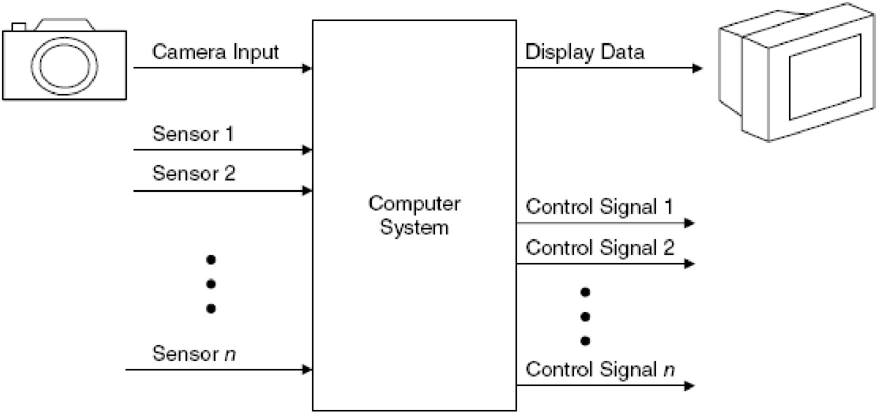
Dependability Attributes



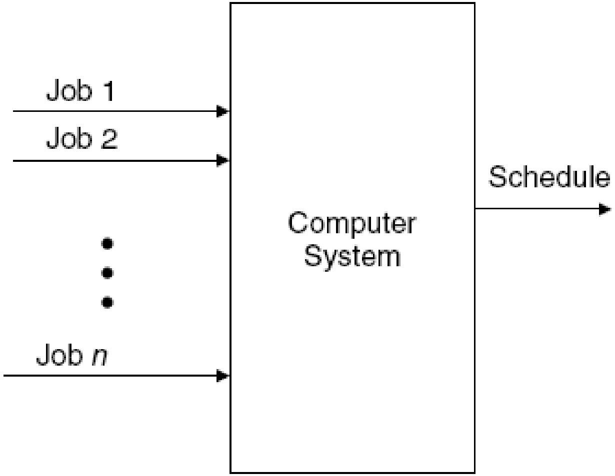
Relationship between Dependability and its Impairments, Means and Measures



A Typical Real-Time Control System

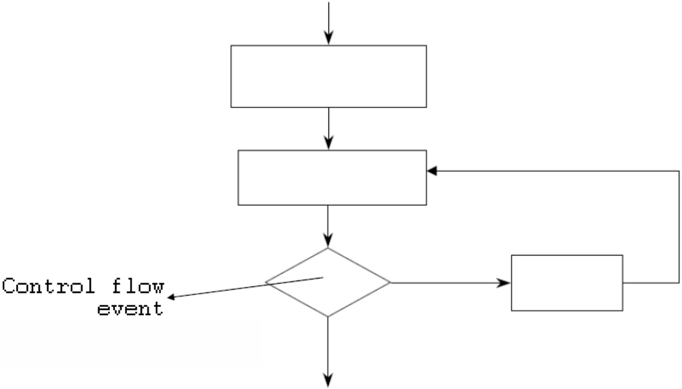


Representation of RT Systems



Events

Any occurrence that causes the program counter to change non-sequentially is considered a change of flow-of-control, and thus an event.



Synchronous vs. Asynchronous Events

	Periodic	Aperiodic	Sporadic
Synchronous	Cyclic code	Typical branch instruction	Branch instruction, e. g. error recovery
	Process scheduled by internal clock	Garbage collection	Traps
Asynchronous	Clock-generated interrupt	Regular but not fixed-period interrupt	Externally generated exception
			Random events

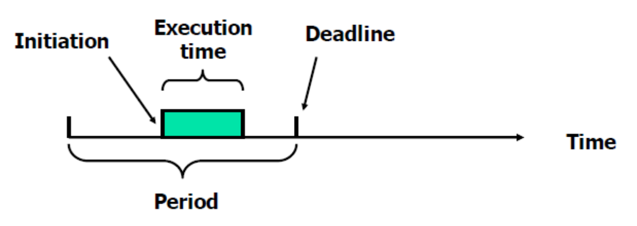
Measuring System Performance

CPU Utilisation

Let C be the execution time and T be the period of a task. Then the utilisation U is defined as

$$U = \sum_{i=i}^{n} \frac{C_i}{T_i}$$

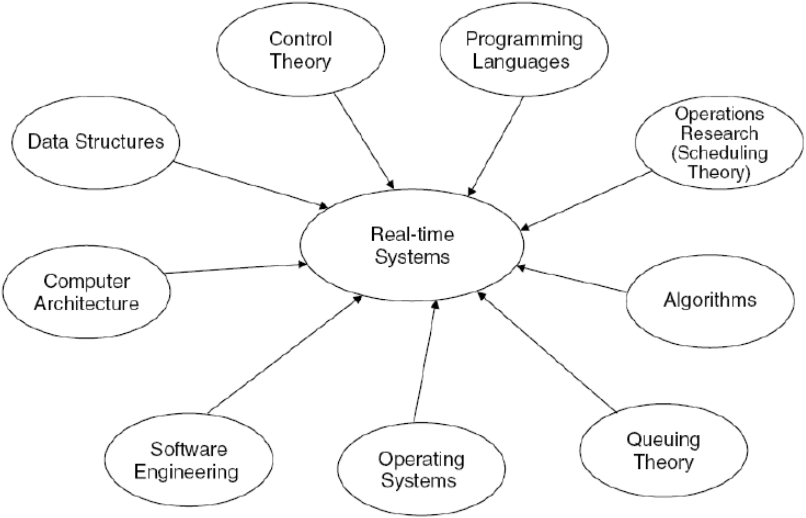
Task Model



CPU Utilization Zones

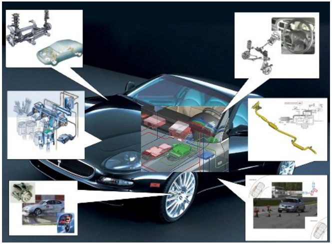
Utilisation (%)	Zone Type	Typical Application
0–25	Significant excess processing power; CPU Power is more powerful than necessary	Various
26–50	Very safe	Various
51–68	Safe	Various
69	Theoretical limit	Embedded systems
72–82	Questionable	Embedded systems
83–99	Dangerous	Embedded systems
100+	Overload	Stressed system

Disciplines that have impact on RT Systems



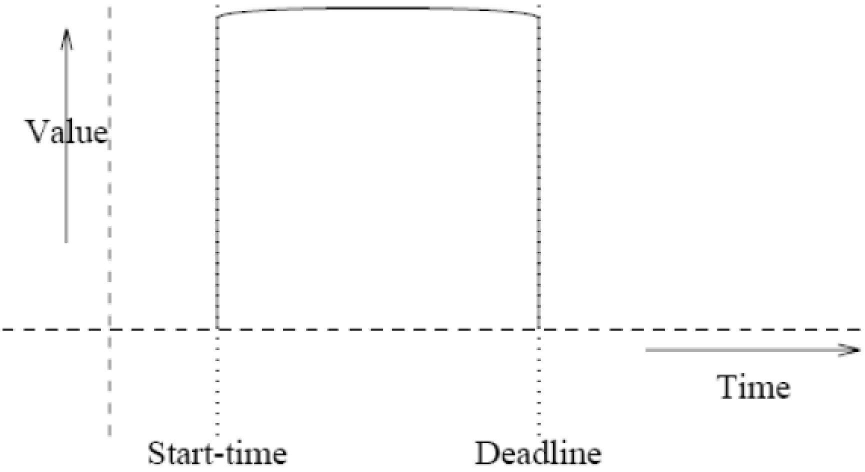
Domains and Applications of RT Systems

Domain	Applications
Avionics	Navigation Displays
Multimedia	Games Simulators
Medicine	Robot surgery Remote surgery Medical imaging
Industrial Sy-stems	Robotic assemb-ly lines Automated in-spection
Civilian	Elevator control Automotive sy-stems

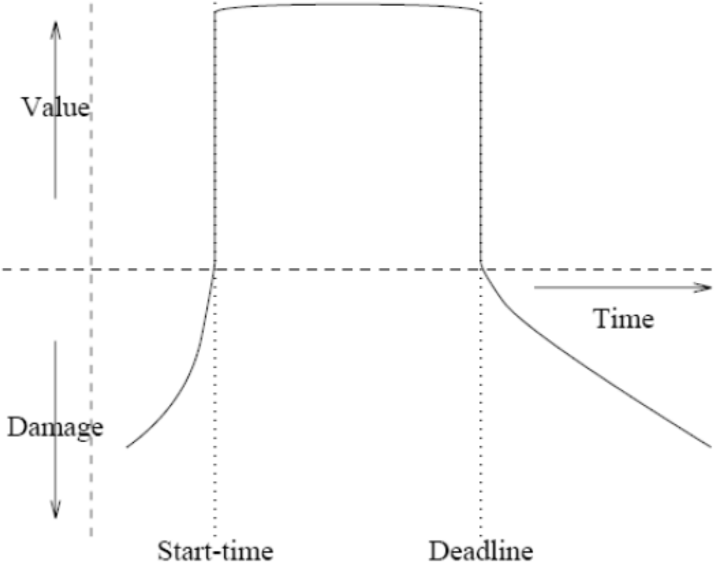


Lerneinheit Einführung

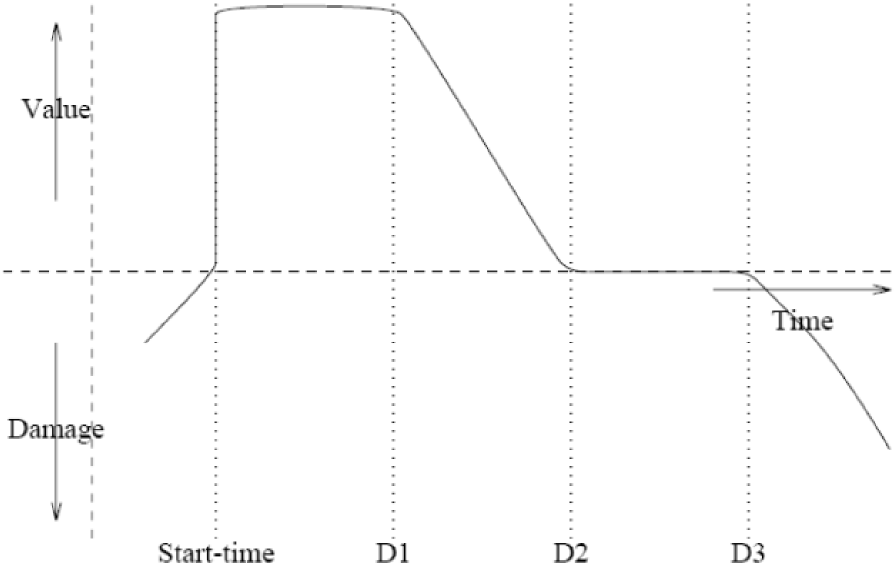
Hard Deadline



Safety Critical System



Hybrid System



Lerneinheit Einführung

Summary

- A real-time system is deterministic (in events an time)
- There are hard, firm and soft real-time systems
- Real-time systems are often embedded systems
- Dependability is a main requirement for real-time systems
- CPU utilisation is a common measure for real-time systems