

Eigenschaften mobiler und eingebetteter Systeme:

Einführung

EMES

Dr.-Ing. Matthias Werner
Dipl.-Inf. Jan Richling
Wintersemester 2001/2002

Regulaarien

- EMES – Eigenschaften mobiler und eingebetteter Systeme
 - Dozenten: Matthias Werner & Jan Richling
- Ort & Zeit:
 - Mo, 11-13 Uhr, Raum 4.111
 - Di, 13-15 Uhr, Raum 4.110
- Begleitendes Projekt
 - Einführung am 23.10.2001

Prüfung

- Wertung als Halbkurs in der Technischen Informatik
- Mündliche Prüfung
- Voraussetzung für die Prüfungszulassung ist eine erfolgreiche Projektverteidigung

Begriff

Was sind mobile und eingebettete Systeme?

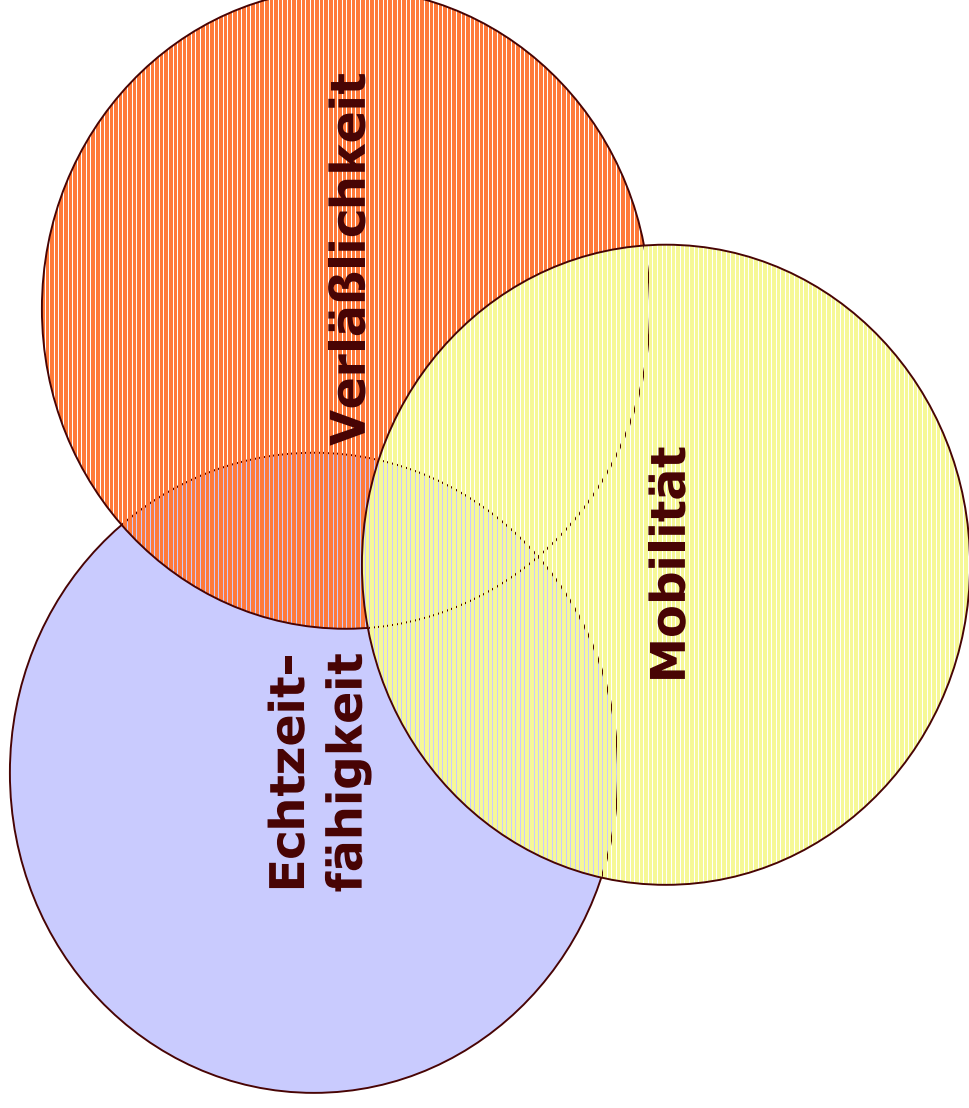
- Auch hier: Prozessoren, die Informationen einlesen, verarbeiten und ausgeben → Standard
- Begriff intuitiv klar, aber schwer abgrenzbar
- Beispiele
 - Steuerrechner
 - PDAs
 - Mobiltelefone

Mobile und eingebettete Systeme

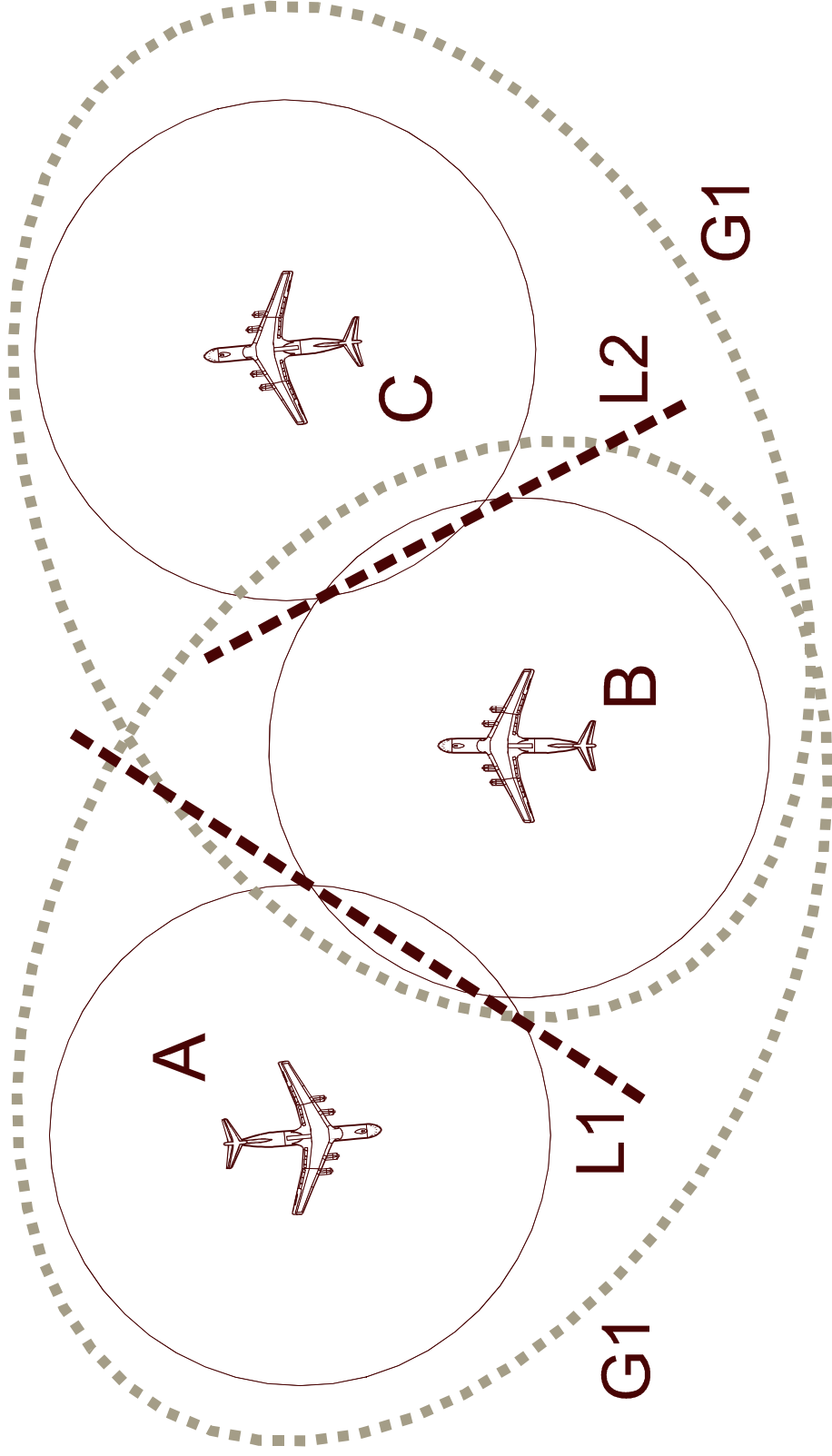
- Was ist der Unterschied zwischen MES und Standardrechnern? (COTS = *commercial of the shelf*)

	COTS	MES
<i>Verlässlichkeit</i>	Niedrig	Hoch
<i>Zeitverhalten</i>	Egal	Vorhersagbar
<i>Umgebung</i>	Standardisiert	Speziell
<i>Nutzerkommunikation</i>	Flexibel	Eingeschränkt
<i>Nutzerinterfaces</i>	Standardisiert	Speziell
<i>Ressourcen</i>	Viel	Wenig
<i>Gruppenbildung</i>	Starr	Flexibel

Drei Eigenschaften von MES



Beispiel



Funktionale vs. nichtfunktionale Eigenschaften

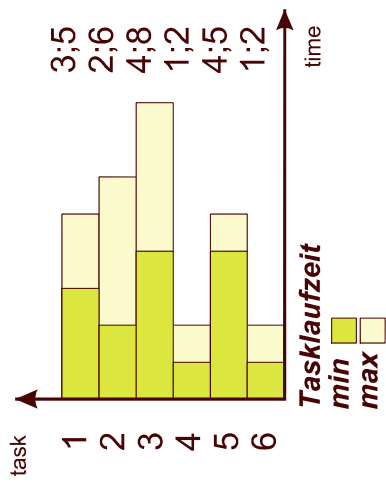
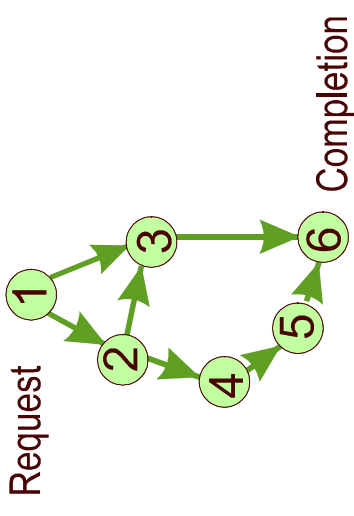
- Üblicherweise interessiert die eigentliche Funktion oder deren Ergebniswert
 - Programmiersprachen
 - Interfacebeschreibungen
- Bei nichtfunktionalen Eigenschaften spielen „Randaspekte“ eine Rolle
 - Ausführungszeit
 - Ressourcenverbrauch
 - Verlässlichkeitsparameter
 - Seiteneffekte
 - Sicherheit

Probleme mit nichtfunktionalen Eigenschaften

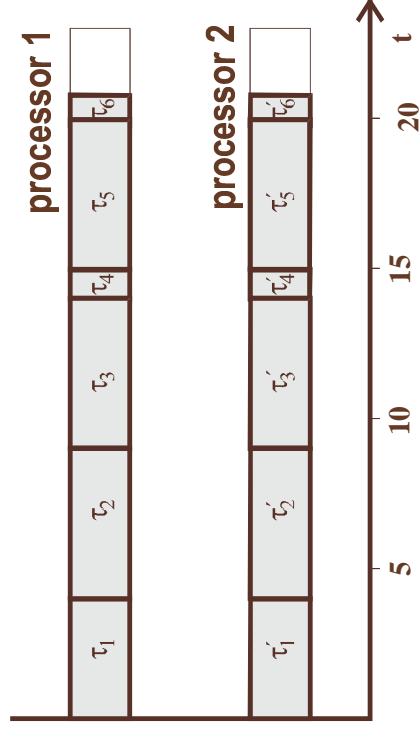
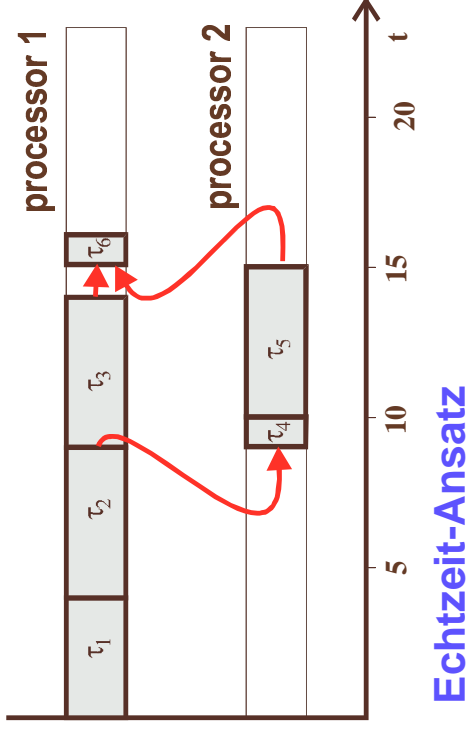
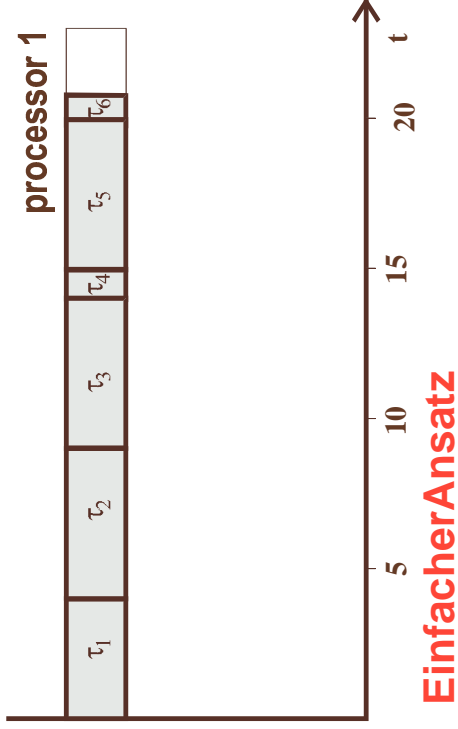
- Definitionen und Maße sind schwierig
- Divide and conquer (Teile und herrsche)
funktioniert häufig nicht
- Abhängigkeit zwischen verschiedenen
nichtfunktionalen Eigenschaften
- Vielfach probabilistische Abhängigkeiten

Abhängigkeiten der Eigenschaften

- Fallstudie:
 - Dienst hängt von der Ausführung von 6 Tasks ab
 - Beendigungswahrscheinlichkeit jeder Task ist in einem gewissen Intervall gleichverteilt
 - ein oder zwei Prozessoren stehen zur Verfügung
 - Deadline 25
 - Fehlerrate $\lambda=0,01$; $R(t)=e^{-\lambda t}$
 - Sollen ein oder zwei Prozessoren eingesetzt werden?
 - Welche Schedulingstrategie soll genutzt werden?



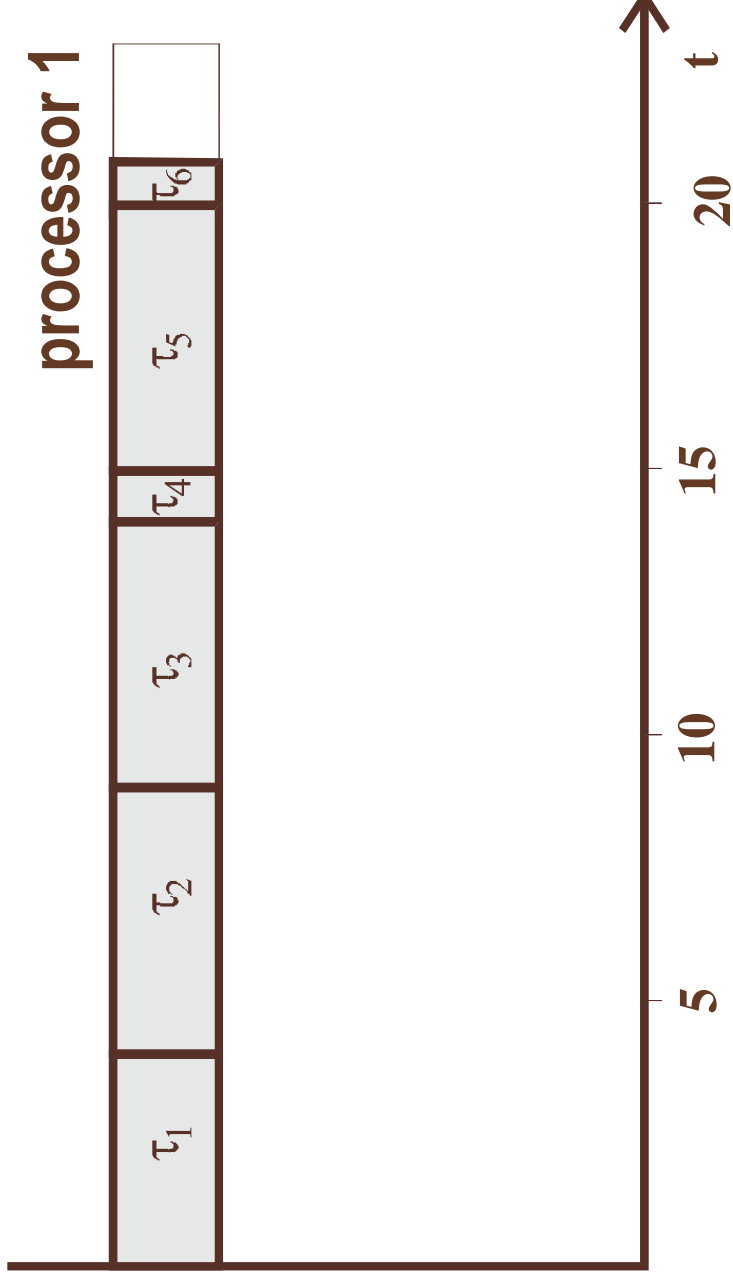
Alternative Designs



Fehlertoleranter Ansatz

Adaptiver Ansatz

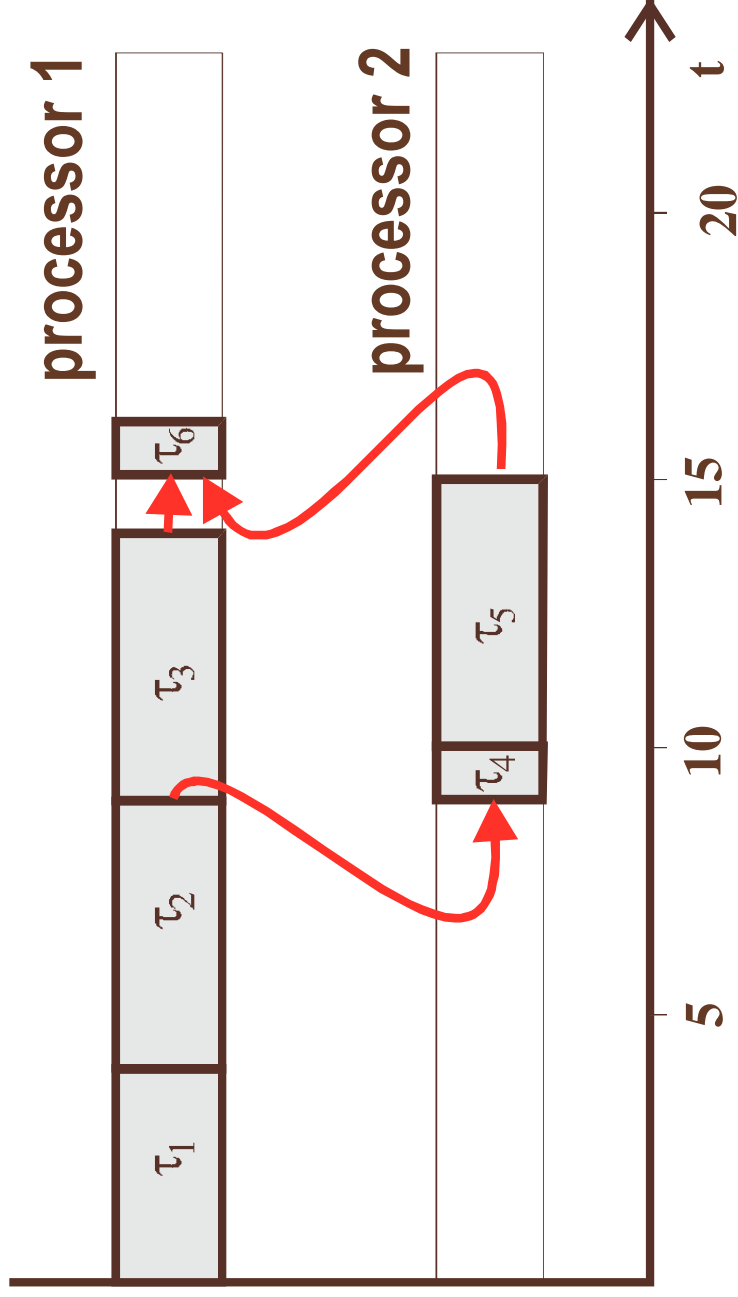
Alternative Designs



Design I: Einfacher Ansatz

- Ein Prozessor, sequentielle Ausführung
- Keine Fehlertoleranz, keine Echtzeit

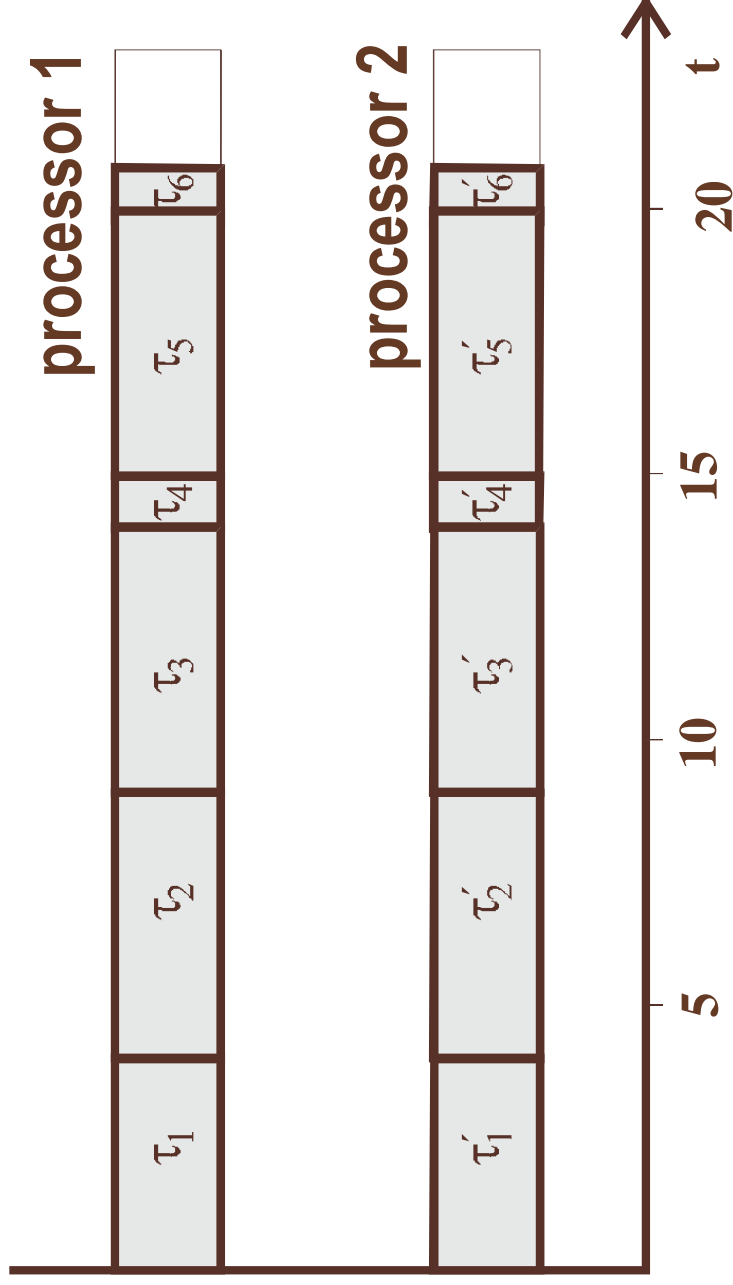
Alternative Designs



Design II: Echtzeit-Ansatz

- Zwei Prozessoren, verteilte Ausführung
- Keine Fehlertoleranz, aber Echtzeit-Garantie

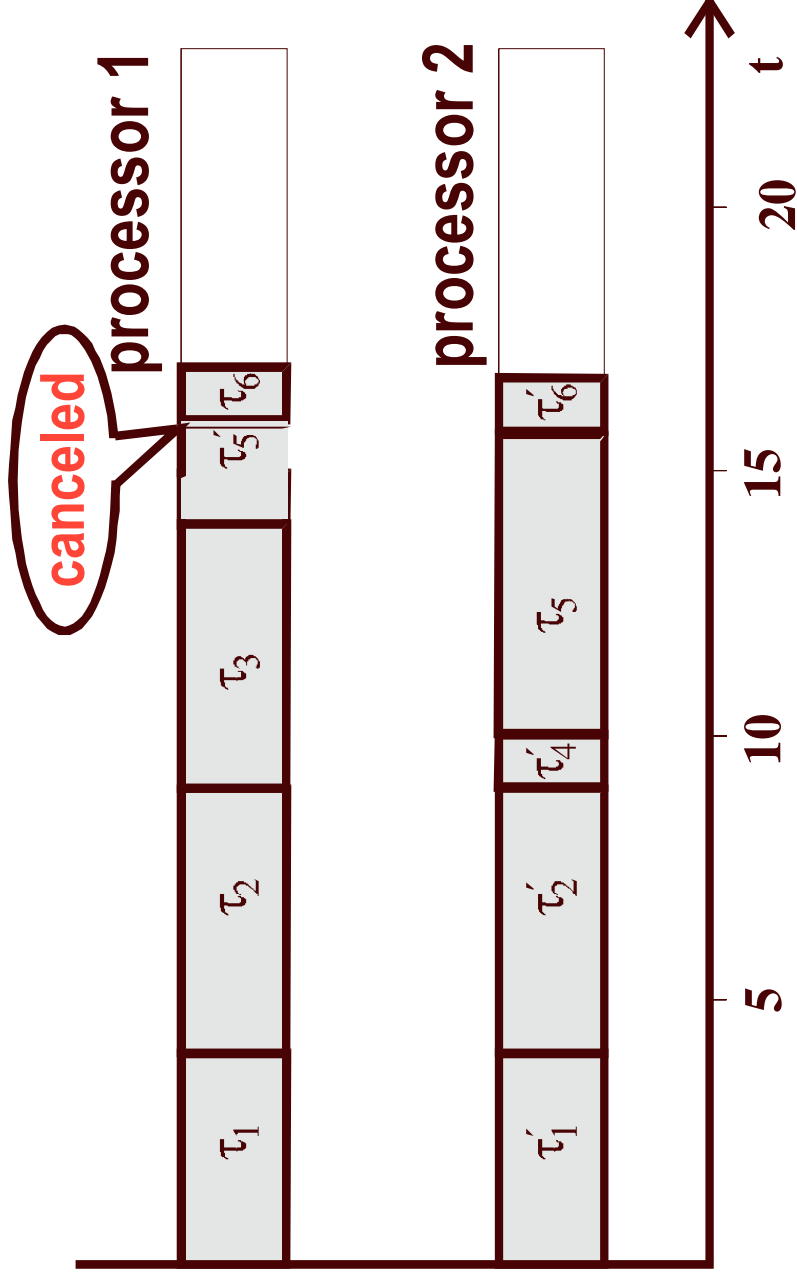
Alternative Designs



Design III: Fehlertoleranter Ansatz

- Zwei Prozessoren, redundante Ausführung
- Fehlertoleranz, aber keine Echtzeit-Garantie

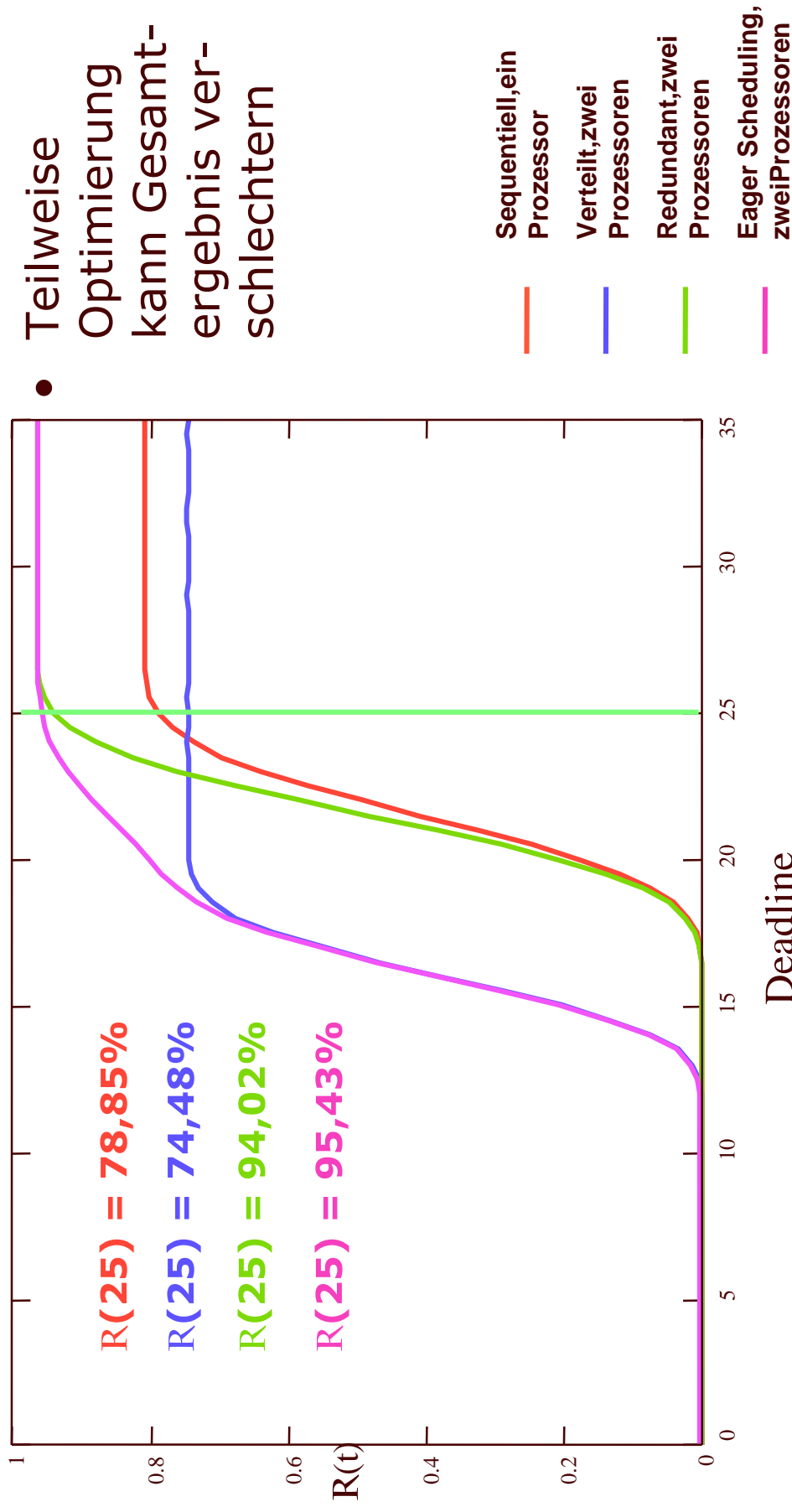
Alternative Designs



Design IV: Adaptiver Ansatz

- Zwei Prozessoren, Eager-Scheduling
- Fehlertoleranz und Echtzeit je nach Situation

Vergleich



Konzept von EMES

- JoJo-Ansatz:
 - Sowohl theoretische Probleme als auch technische Anwendungen im Wechsel
 - Praktische Erfahrungen durch Projektarbeit
- Vier Module (z.T. verschränkt):
 - Echtzeit
 - Verlässlichkeit
 - Kommunikation
 - Systeme & Fallstudien

(Vorläufige) Themenplanung

- GrundlagenEchtzeit:
Harte/weiche,Scheduling,
Abhängigkeiten, Taskmodell
- RMA/EDF
- WeitereProblemedesRT -
Schedulings
- WeicheEchtzeit:Multimedia
- Regelungen
- RT-Anwendungen
- Verlässlichkeitskonzepte
- FaultDiagnosis/ Byzantine
Agreement
- Fehlertoleranzund
Recoverytechniken
- DrahtloseKommunikation
- Standardsmobilen
Kommunikation(GSM,
UMTS,etc.)
- Feldbusse
- Fehlertolerante
Kommunikation
- Echtzeitkommunikation
- Gruppenkommunikation
- Uhrensynchronisation
- Embedded &RT -
Betriebssysteme(Win CE,
RT-Linux,OSEC/Pure)
- Architekturen:TTAundMSS
- Modellierung

Literatur I

- J.W.S. Lui: *Real-Time Systems*, Prentice Hall, 2000
- C.M. Krishna, K.G. Shin: *Real-Time Systems*, McGraw-Hill, 1997
- H. Kopetz: *Real-Time Systems: Design Principles for Distributed Embedded Applications*, Kluwer Academic, 1997
- T. Anderson, P.A.Lee: *Fault Tolerance Principles and Practice*, Prentice Hall 1982
- D.K. Pradhan (Hrsg.): *Fault Tolerant Computer System Design*, Prentice Hall PTR, 1996
- G.T. Coulouris, J. Dollimore, T. Kindberg: *Distributed Systems*, Addison-Wesley, 1994

Literatur II

- H. Unbehauen: *Regelungstechnik I+II*, Vieweg, 1989
- Semesterapparat EMES
- Foliensatz