

EMES: Eigenschaften mobiler und eingebetteter Systeme

Echtzeitsysteme: Grundlagen

Dr. Felix Salfner, Dr. Siegmund Sommer
Wintersemester 2010/2011



Was ist Echtzeit?

- Es gibt eine Reihe verwirrender Vorstellungen, was ein Echtzeitsystem ist, beziehungsweise welche Eigenschaften ein solches ausmachen

Was ist Echtzeit?

- Es gibt eine Reihe verwirrender Vorstellungen, was ein Echtzeitsystem ist, beziehungsweise welche Eigenschaften ein solches ausmachen
- Echtzeitsysteme müssen
 - “schnell” sein?
 - vorhersagbar sein?
 - rechtzeitig Ergebnisse liefern?
 - “sicher” sein?
 - fehlerfrei arbeiten?
 - so konstruiert sein, daß sie auch im Fehlerfall noch funktionieren?
- Definition ist erforderlich

Was ist Echtzeit?

finanzen.net

Name, WKN, ISIN > KURS
 Realtime

Dax 5.731 -1,8% Dow 9.949 -0,9% Euro 1.4957 -0,3%
 E50 2.888 -1,7% Nas 2.151 -0,6% Öl 78,63 -1,0%
 TDax 760 -2,1% NasF 1.669 -1,0% Gold 1.055 -0,3%

Aktien | Zertifikate & OS | Fonds | ETFs | Anleihen | Zinsen | News | Analysen | Rohstoffe | Devisen | Forum | Priv. Finanzen | myfinanzen

Aktienkurse | Suche | Portfolio | Realtimekurse | Indizes | Listen | Termine | IPO | Insiderdaten | Dividenden | Chartanalyse | Ticker | Jobs

DAX

Kurse	Charts	News	Listen	Statistik	Info	zugeh. Wertpapiere
Snapshot	Chart (groß)	Marktberichte	enthaltene Werte	Performance	HV-Termine	Zertifikate
Historisch	Chartvergleich	Analysen	Top/Flop	Hoch/Tief	Dividenden	Optionsscheine
Realtime Liste	Seit 1959	RSS-Feed	Volumen	Korr-Matrix		Knock-outs
Parkettkamera	Quickcharts		Marktkapitalisierung	DAX-Liga		
	Realtime Chart					

DAX DB Realtime Indikation

5.731,00 Punkte			
Kurszeit	11:39:50	Vortag	5.833,49
+/-	-.102,49	%	-1,76%

- Anstieg gegenüber vorhergehendem Kurs
- Rückgang gegenüber vorhergehendem Kurs

Optionen

- Performance-Daten einblenden

Realtime-Kurse:

Boerse Stuttgart

WKN / ISIN



Die Börse Stuttgart bietet Ihnen täglich zwischen 9:00 und 20:00 Realtime Bid- und Ask-Kurse zu allen an der Börse Stuttgart gehandelten

Wertpapieren.

Realtime-Kurse außerbörslich:

LANG & SCHWARZ

WKN



Lang & Schwarz "Trade-Link" bietet Ihnen außerbörsliche Realtime-Kurse in der Zeit von wochentags 08:00 bis 23:00, Samstags 10:00 bis 13:00 und Sonntags 17:00 bis 19:00.

Was ist Echtzeit?

MAXIM
INNOVATION DELIVERED

Enter keywords or part number

What's New Products Solutions Design AppNotes Support Sales About Us Me

Maxim > Products > Real-Time Clocks

Real-Time Clocks (RTCs)

If it's electronic, it needs a clock...

Maxim real-time clocks (RTCs) are used in dozens of applications. Their features include time-of-day, calendar, trickle charger, and memory functions. Interface options include Bus, Byte-wide, Phantom, I²C, 3-Wire, and SPI™ Bus. Our family of precision time keep devices offer extreme accuracy and RTC functionality.

- Consumer Electronics
- Industrial Controls
- Telecommunications
- Office Equipment
- Medical
- Utility Power Meters
- Test and Measurement
- GPS

RTC Homepage

Maxim RTC Technology

- Integrated Crystal Package

Product Information

- Latest Data Sheets
- Cross Reference Search
- RoHS/Lead-Free Lookup
- Weekly New Product Updates
- Didn't Find What You Need?

Design Help

- Application Notes
- Real-Time Clocks Design Guide (PDF)
- System Timing and Control Design Guide (PDF)
- Evaluation Kits
- Real-Time Clock Calculator
- Super Capacitor Calculator

Support

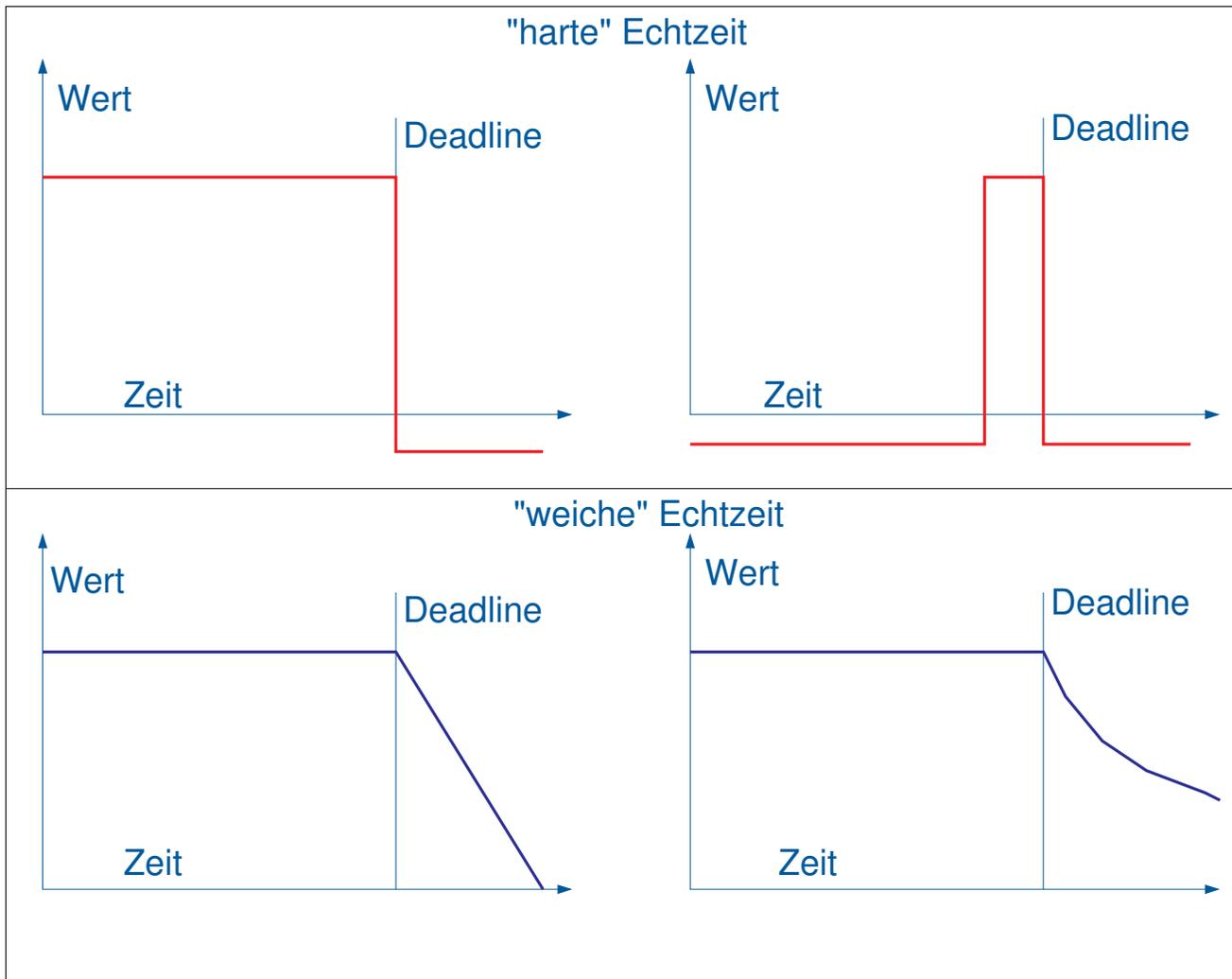
- Samples
- Buy

Definition 1 - Krishna und Shin

“Any system where a timely response by the computer to external stimuli is vital is a real-time system”

- “Rechtzeitig” — definiert über Deadlines
- Deadlines — spätester Zeitpunkt der Vollendung einer Task
- rechtzeitige Reaktion — immer? manchmal? in den meisten Fällen?
- Ist dann jeder Computer ein Echtzeitsystem?

Definition 2 - Resultat / Wert-Funktion



Echtzeitsysteme sind Computersysteme, bei denen der Nutzen eines Resultates nicht nur vom Resultat abhängt, sondern auch vom Zeitpunkt der Auslieferung des Resultats.

Definition 3 - analog Krishna und Shin

Ein Echtzeitsystem ist alles, was wir in dieser Vorlesung als Echtzeitsystem verstehen. Das umfaßt eingebettete Systeme wie beispielsweise

- Steuerungen in Fahr- und Flugzeugen
- Automatisierungsanlagen in der Industrie und in Kraftwerken
- ... und andere Objekte, wo Unglücke geschehen, wenn das Computersystem nicht rechtzeitig Ergebnisse liefert

Diese System nennen wir *harte* Echtzeitsysteme.

Im Gegensatz dazu gibt es auch *weiche* Echtzeitsysteme wie Multimedia-Systeme, bei denen verpaßte Deadlines “nur” die Qualität des Ergebnisses beeinträchtigen, aber keine weiteren Auswirkungen haben.

Beispiel für Steuerungen: Auto und Fahrer

- Fahrer:
Echtzeitsteuerung
- Auto:
Gesteuerter Prozeß
- Bedienelemente:
Aktuatoren
- Mission:
Von A nach B fahren ohne Kollisionen mit anderen Fahrzeugen,
Personen oder Gegenständen und unter Einhaltung der Verkehrsregeln

Performance-**maße** I

Wie mißt man die “Performance” des Fahrers? Genügt “Ankommen”?

- Maß: Zeit bis zum Ankommen
 - Mit 15 km/h sicher fahren in einem Schneesturm ist ein akzeptables Ergebnis
 - Mit 15 km/h auf einer leeren Autobahn fahren ist inakzeptabel
- Maß: Unfallfreiheit
 - In den Graben fahren, weil das Auto in einer Kurve ins Schleudern gekommen ist, ist inakzeptabel
 - In den Graben fahren, um damit einen (unverschuldeten) Frontalzusammenstoß zu vermeiden, ist akzeptabel

Performance-Maße hängen von der Umgebung ab und können nur relativ zu dem bestmöglichen Ergebnis für die jeweiligen Parameter der Umgebung gemessen werden.

Performancemaße II

“Der Fahrer war während 99.9% der Fahrzeit wach”

→ Nutzlose Information: 0.1% einer achtstündigen Fahrt sind 28 Sekunden

“Der Fahrer hat nie länger als 500 ms geschlafen”

→ Verwertbare Information (unter gewissen Nebenbedingungen - ein Fahrzeug legt bei 200 km/h in dieser Zeit fast 28 Meter zurück)

- Durchschnittswerte sind nicht ausreichend
- Werte sind von Nebenbedingungen abhängig

Tasks und Deadlines

- Was sind zeitkritische Tasks?
- Was sind ihre Deadlines?
- Was sind unkritische Tasks?

Tasks und Deadlines

- Was sind zeitkritische Tasks?
 - Bremsen
 - Lenken
 - (Beschleunigen)
- Was sind ihre Deadlines?
- Was sind unkritische Tasks?

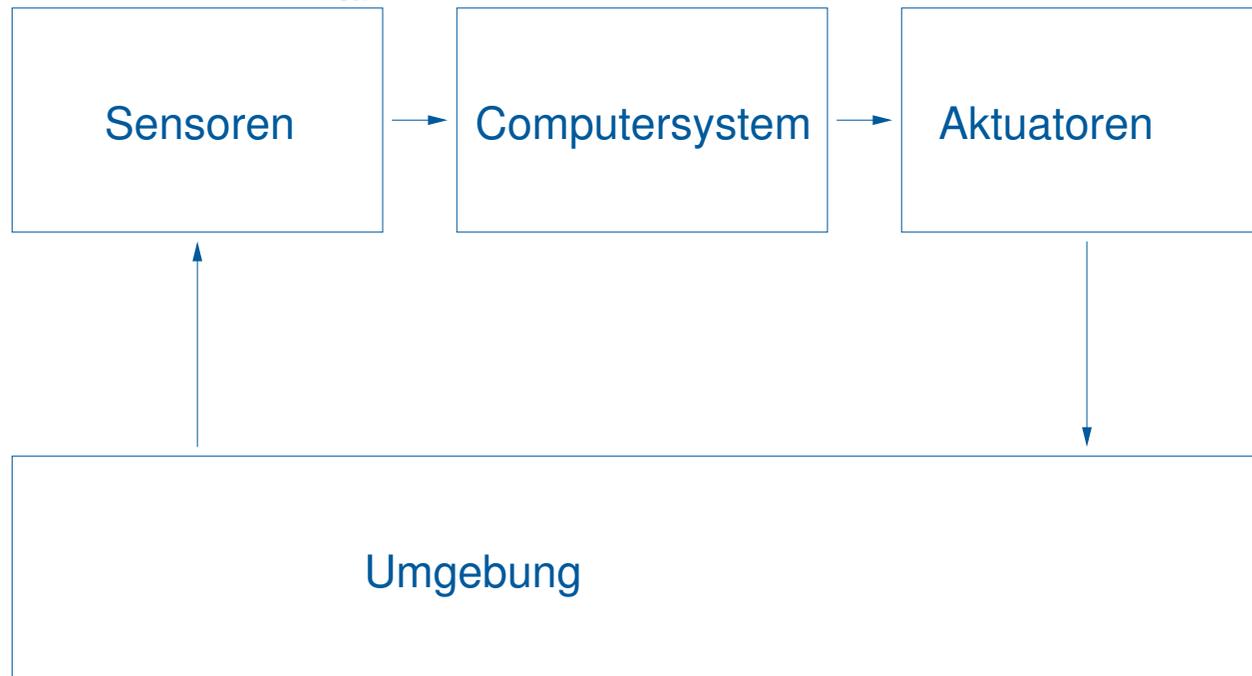
Tasks und Deadlines

- Was sind zeitkritische Tasks?
 - Bremsen
 - Lenken
 - (Beschleunigen)
- Was sind ihre Deadlines?
 - Hängen von der Situation ab:
 - * Sonntag, 6 Uhr morgens, auf leerer Straße
 - * Werktags, 16 Uhr, Innenstadtverkehr
 - Task-Deadlines in Echtzeitsystemen sind nicht konstant
- Was sind unkritische Tasks?

Tasks und Deadlines

- Was sind zeitkritische Tasks?
 - Bremsen
 - Lenken
 - (Beschleunigen)
- Was sind ihre Deadlines?
 - Hängen von der Situation ab:
 - * Sonntag, 6 Uhr morgens, auf leerer Straße
 - * Werktags, 16 Uhr, Innenstadtverkehr
 - Task-Deadlines in Echtzeitsystemen sind nicht konstant
- Was sind unkritische Tasks?
 - Radio anschalten
 - Heizung bedienen
 - Licht einschalten
 - Aber! Auch diese Tasks können “weiche” Deadlines haben.

Struktur einer Echtzeitanwendung



- Kann auch zusätzlich Operator beinhalten
 - Werden Sensorwerte und Systemaktionen angezeigt
 - Reagiert mit paralleler Beeinflussung der Aktuatoren

Charakterisierung von Tasks I

Nach ihrer Wichtigkeit

- kritische Tasks

Verpassen einer Deadline hat katastrophale Auswirkungen auf die Echtzeitanwendung

Ziel: Alle Deadlines müssen eingehalten werden

- unkritische Tasks

Verpassen von Deadlines ist unkritisch.

Ziel: Möglichst viele unkritische Tasks innerhalb der Deadline beenden

Charakterisierung von Tasks II

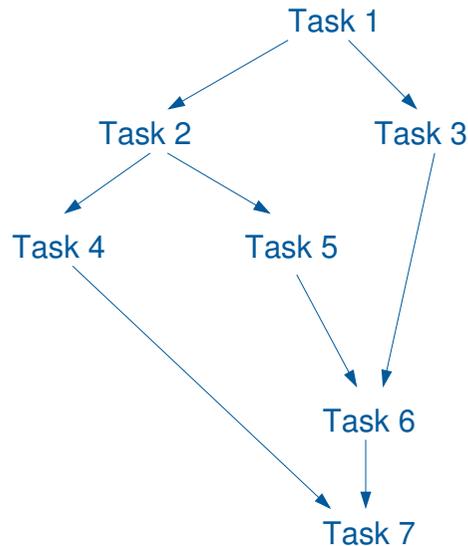
Nach dem Zeitpunkt ihres Auftretens

- periodische Tasks
Task wird periodisch alle x Zeiteinheiten (*Periode*) wiederholt ausgeführt
Beispiel: Steuerungstasks
- aperiodische Tasks
Task tritt unregelmäßig und unvorhersagbar auf
Beispiel: Alarmierungstasks
Spezialfall: Sporadische Tasks — zeitlicher Abstand (*minimal interarrival time*) zwischen zwei Auftreten ist begrenzt

Charakterisierung von Tasks III

Nach Abhängigkeiten von anderen Tasks

- Unabhängige Tasks
können in jeder beliebigen Reihenfolge ausgeführt werden
- Abhängige Tasks
Abhängigkeiten sind in einem Precedencegraphen darstellbar als
“vorher”-Relation

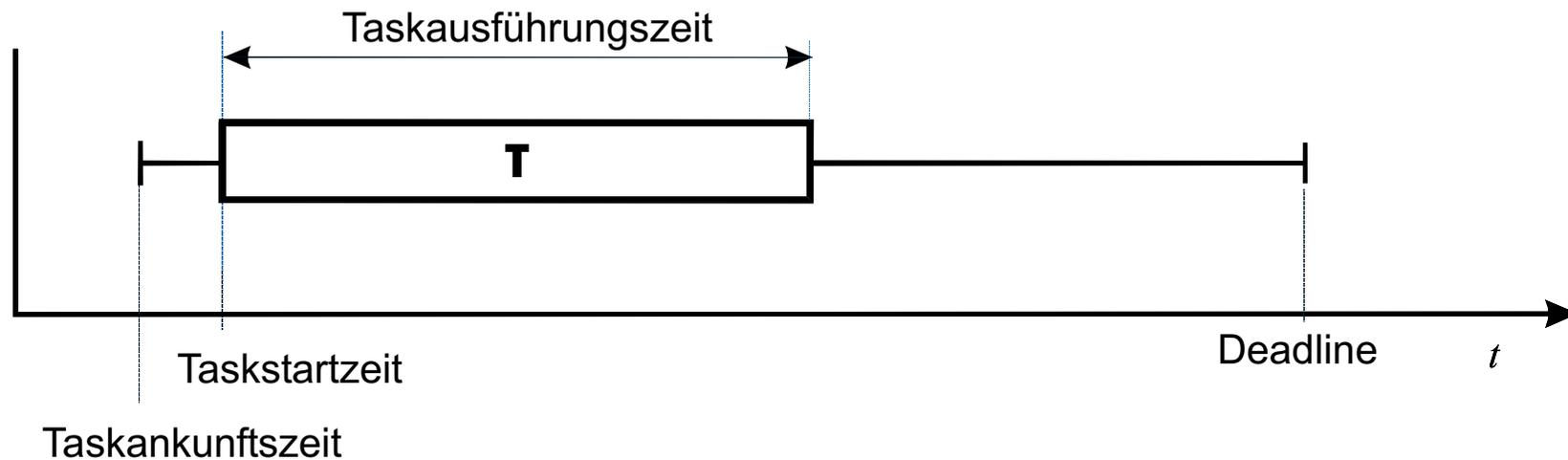


Periodisches Taskmodell

- Motivation: Reduzierung der Komplexität durch vereinfachende Annahmen
- Ausgangsidee: Viele Echtzeitanwendungen erfordern periodische Tasks
- Annahme: Tasks sind unabhängig
- Vorteil: Annahme von Periodizität und Unabhängigkeit führt zu Wissen über die zukünftige Belastung des Systems
- Problem: Es gibt auch aperiodische Tasks, die behandelt werden müssen!

Parameter einer Task I

- Taskankunft r_i
- Taskstartzeit
- Ausführungszeit e_i
- Deadline D_i
- Periode P_i
- Maß für “Wichtigkeit” (Priorität)



Parameter einer Task II



- Auslastung (Utilization): $U_i = \frac{e_i}{P_i}$
Bemerkung: $\forall i : U_i \leq 1$
- Jitter
 - exakte Periodizität ist real nicht immer erreichbar
 - Schwankungen der Ankunftszeit heissen *Jitter*
- Unterbrechbarkeit
Darf die Abarbeitung der Task unterbrochen werden?

Behandlung von aperiodischen und sporadischen Tasks

- Minimale Zeit zwischen der Ankünften der Task muß bekannt sein (*minimal interarrival time*)
- Task wird im Modell als periodisch mit einer Periode entsprechend der minimal interarrival time angenommen
- Nachteil: Ausführung findet nicht in jeder Periode statt, damit wird Last als höher angenommen, als sie ist
- Vorteil: Periodisches Taskmodell ist anwendbar, geringes Vorwissen über die Task genügt
- Nur anwendbar auf den Spezialfall einer sporadischen Task, nicht allgemein für aperiodische Tasks

Generelles Problem: Aufteilung von Ressourcen an konkurrierende “Verbraucher”

Ressourcen bei Echtzeitsystemen (u.a.):

- CPU-Zeit (klassisches Schedulingproblem)
- Bandbreite von Netzwerkanbindungen
- I/O (Zugriff auf Geräte)
- Energie



CPU-Scheduling

klassische (nicht Echtzeit) Ziele:

- Hohe Auslastung des Systems
- Fairness
- keine Starvation (lang anhaltende Verdrängung einzelner Tasks)
- Antwortverhalten (kurze Reaktionszeit)
- hoher Durchsatz (abgearbeitete Prozesse pro Zeiteinheit)

→ So nicht anwendbar bei Echtzeitsystemen!

Schedulingstrategien für Nicht-Echtzeit I

- First come first serve (FCFS)
 - Prozesse werden in der Reihenfolge ihrer Ankunft abgearbeitet
 - Fair, effizient bei geringer Last
 - Ungünstigster Fall: Lange Task kommt als Erstes an
- Shortest job first (SJF)
 - Wissen über Laufzeit der Prozesse
 - Prozeß kommt - entsprechend der Laufzeit in Liste eingeordnet
 - auch als unterbrechende Version (shortest remaining time first - SRTF)
 - gute Antwortzeiten

Beispiel SJF

- Drei sequentiell eintreffende Tasks: $e_1 = 10, e_2 = 1, e_3 = 1$
- Durchschnittliche Wartezeit mit FCFS: 7
- Durchschnittliche Wartezeit mit SJF: 1
- FCFS ist fair, kann aber zu einer schlechteren durchschnittlichen Wartezeit führen
- SJF kann besseren Durchsatz bewirken, aber Gefahr von Starvation
- Wissen über Laufzeit meist nicht vorhanden

Schedulingstrategien für Nicht-Echtzeit



- Round robin (RR)
 - Prozesse laufen ohne Unterbrechung für eine bestimmte Zeitdauer (*quantum*), gehen dann an das Ende der ready-Liste
 - Hoher Durchsatz bei Tasks mit $e_i \leq \text{quantum}$
 - FCFS mit Unterbrechung (gleiches Verhalten für grosses Quantum)
- Prioritätsscheduling
 - Eine ready-Liste pro Priorität, absteigend sortiert
 - statisch: Prioritäten werden einmal gesetzt und bleiben gleich
 - dynamisch: Prioritäten werden zur Laufzeit modifiziert
 - Priority Aging → lange wartende Prozesse werden temporär auf eine höhere Priorität gesetzt

- Tasks: $e_A = 8, e_B = 4, e_C = 9, e_D = 5$

Metrik	FCFS	SJF	RR
Durchschnittliche Wartezeit (Zeit in der ready-Liste)	10.25	7.5	15
Durchschnittliche Antwortzeit (Zeit bis zur ersten Ausführung)	10.25	7.5	4.5
Durchschnittliche 'Turnaround'-Zeit (Ankunft bis Fertigstellung)	16.75	14	21.5

- Zusätzliche Metriken Durchsatz und Auslastung
→ Bei Nicht-Echtzeit primär sinnvoll in MP-Systemen

Echtzeitscheduling I

Verfahren des Nicht-Echtzeitscheduling können nicht übernommen werden:

- Deadlines müssen erfüllt werden
- Tasks sind verschieden wichtig → Keine Fairness
- Kurze Reaktionszeiten genügen nicht, Zeiten müssen garantiert sein
- Weitere Parameter sind wichtig:
 - Deadline
 - Periode
 - Abhängigkeiten von anderen Tasks (Taskgraph)

Echtzeitscheduling II

Schedule kann erzeugt werden:

- Vor Laufzeit des Echtzeitsystems: *offline-scheduling*
 - unflexibel gegenüber Änderungen
 - maximale Auslastung stets erreichbar
 - kaum Aufwand zur Laufzeit
- Während der Laufzeit des Systems: *online-scheduling*
 - flexibel gegenüber Änderungen
 - erreichbare Auslastung abhängig vom Verfahren
 - Aufwand zur Laufzeit höher
 - meist prioritätenbasiert

Prioritätenbasiertes Echtzeitscheduling

Zuweisung der Prioritäten erfolgt:

- statisch (keine Änderung während des Ablaufes)
 - Einfache und einmalige Berechnung bei Taskankunft
 - Festlegung anhand statischer Parameter der Task, beispielsweise Periodenlänge
 - Beispiel: RMS (*rate monotonic scheduling*)
- dynamisch
 - dynamische Neuberechnung der Prioritäten
 - Festlegung in Abhängigkeit von dynamischen Parametern der Task wie der aktuellen Deadline
 - Beispiel: EDF (*earliest deadline first*)