

# Real-Time Scheduling

03. Dezember 2008

Einführung

**Problemstellung**

Task Charakteristiken

Task Dependency

Graph

Notation

Annahmen

RMS

EDF

Probleme der  
besprochenen Ansätze

Task Assignment

Literatur

Einen gültigen Schedule finden, welcher die gegebenen Eigenschaften von Echtzeit-Tasks berücksichtigt.

- Einhaltung der Zeitschranken statt Fairness
- online / offline Scheduling
- Task
  - verbraucht Ressourcen (CPU, Speicher, etc)
  - erzeugt Ergebnis

- Einführung
- Problemstellung
- Task Charakteristiken**
- Task Dependency Graph
- Notation
- Annahmen
- RMS
- EDF
- Probleme der besprochenen Ansätze
- Task Assignment
- Literatur

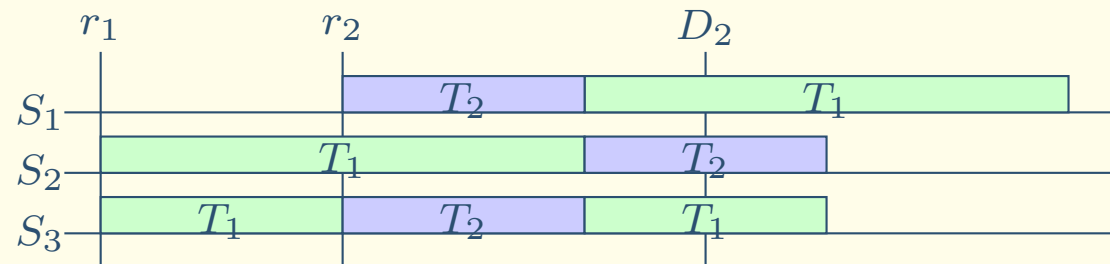
- Einhaltung der Deadline

- critical
- non-critical
- IRIS

- Periodizität

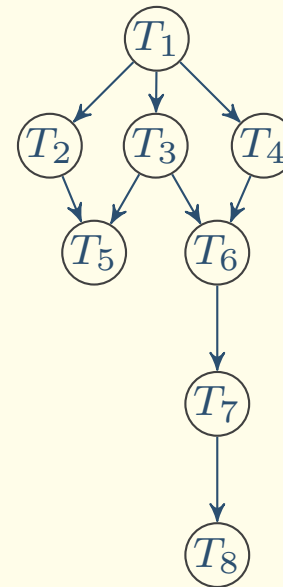
- periodisch
- sporadisch
- aperiodisch

- unterbrechbar



- abhängig von anderen Tasks

- Einführung
- Problemstellung
- Task Charakteristiken
- Task Dependency Graph**
- Notation
- Annahmen
- RMS
- EDF
- Probleme der besprochenen Ansätze
- Task Assignment
- Literatur



$\prec (1) = \emptyset$
$\prec (2) = \{1\}$
$\prec (3) = \{1\}$
$\prec (4) = \{1\}$
$\prec (5) = \{1, 2, 3\}$
$\prec (6) = \{1, 3, 4\}$
$\prec (7) = \{1, 3, 4, 6\}$
$\prec (8) = \{1, 3, 4, 6, 7\}$

- Vorrang Relation  $\prec$

Einführung

Problemstellung

Task Charakteristiken

Task Dependency

Graph

**Notation**

Annahmen

RMS

EDF

Probleme der  
besprochenen Ansätze

Task Assignment

Literatur

- $n$  - Taskanzahl
- Jeweils für Task  $T_i$ :
  - $e_i$  - Ausführungszeit
  - $P_i$  - Periode
  - $r_i$  - Taskankunft
  - $d_i$  - relative Deadline
  - $D_i$  - absolute Deadline
  - $I_i$  - Startzeit der ersten Periode

Einführung

Annahmen

**Annahmen**

RMS

EDF

Probleme der  
besprochenen Ansätze

Task Assignment

Literatur

Alle Tasks im Taskset sind:

- präemptiv
- periodisch
- Deadline = Periode
- voneinander unabhängig

Außerdem:

- Context Wechsel kostet nichts
- nur die Rechenzeit zählt: Speicher, I/O, etc. sind vernachlässigbar

Einführung

Annahmen

RMS

**Rate Monotonic  
Scheduling**

Beispiel

Schedulekriterium I

Schedulekriterium II

Erweiterungen

EDF

Probleme der  
besprochenen Ansätze

Task Assignment

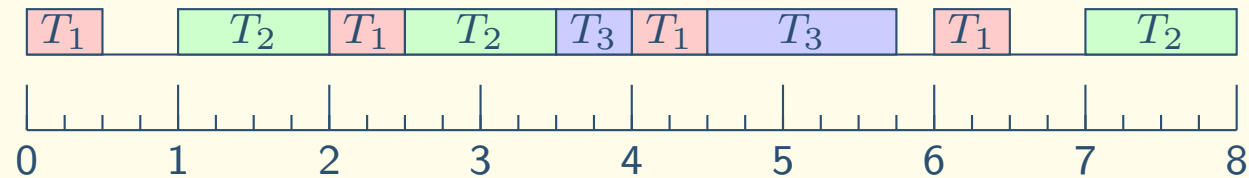
Literatur

- offline Verfahren
- Task Priorität ist statisch und invers zur Periode  
Task mit kürzester Periode unterbricht alle anderen Tasks
- optimaler Algorithmus für feste Prioritäten

- Einführung
- Annahmen
- RMS
- Rate Monotonic Scheduling
- Beispiel**
- Schedulekriterium I
- Schedulekriterium II
- Erweiterungen
- EDF
- Probleme der besprochenen Ansätze
- Task Assignment
- Literatur

Task	$P_i$	$e_i$	$I_i$
$T_1$	2	0.5	0
$T_2$	6	2.0	1
$T_3$	10	1.75	3

- $P_1 < P_2 < P_3$ : Priorität von  $T_1$  am höchsten





Einführung

Annahmen

RMS

Rate Monotonic Scheduling

Beispiel

**Schedulekriterium I**

Schedulekriterium II

Erweiterungen

EDF

Probleme der besprochenen Ansätze

Task Assignment

Literatur

hinreichende Bedingung:

$n$  Tasks können ausgeführt werden, wenn Prozessorauslastung

$$U \leq n * (\sqrt[n]{2} - 1)$$

$n$	1	2	3	5	10	100	$\infty$
$U$	1.0	0.83	0.78	0.74	0.72	0.70	0.69

Beispiel:

Task	$e_i$	$P_i$	$I_i$
$T_1$	40	100	0
$T_2$	40	150	0
$T_3$	100	350	0

- $U = 0.4 + 0.2\bar{6} + 0.\overline{285714} = 0.\overline{952380}$
- $0.\overline{952380} \leq 0.78 \downarrow$

Einführung

Annahmen

RMS

Rate Monotonic  
Scheduling

Beispiel

Schedulekriterium I

**Schedulekriterium II**

Erweiterungen

EDF

Probleme der  
besprochenen Ansätze

Task Assignment

Literatur

Task	$e_i$	$P_i$	$I_i$
$T_1$	40	100	0
$T_2$	40	150	0
$T_3$	100	350	0

notwendige und hinreichende Bedingung:

$T_1$  : Schedule gültig, wenn  $e_1 \leq P_1$

$T_2$  : Schedule gültig, wenn  $e_1 + e_2 \leq P_1 \vee$

$$2 * e_1 + e_2 \leq P_2$$

$T_3$  : Schedule gültig, wenn  $e_1 + e_2 + e_3 \leq P_1 \vee$

$$2 * e_1 + e_2 + e_3 \leq P_2 \vee$$

$$2 * e_1 + 2 * e_2 + e_3 \leq 2 * P_1 \vee$$

$$3 * e_1 + 2 * e_2 + e_3 \leq 2 * P_2 \vee$$

$$4 * e_1 + 3 * e_2 + e_3 \leq P_3$$

Einführung

Annahmen

RMS

Rate Monotonic  
Scheduling

Beispiel

Schedulekriterium I

**Schedulekriterium II**

Erweiterungen

EDF

Probleme der  
besprochenen Ansätze

Task Assignment

Literatur

Task	$e_i$	$P_i$	$I_i$
$T_1$	40	100	0
$T_2$	40	150	0
$T_3$	100	350	0

$$3 * e_1 +$$

$$120 +$$

$$2 * e_2 +$$

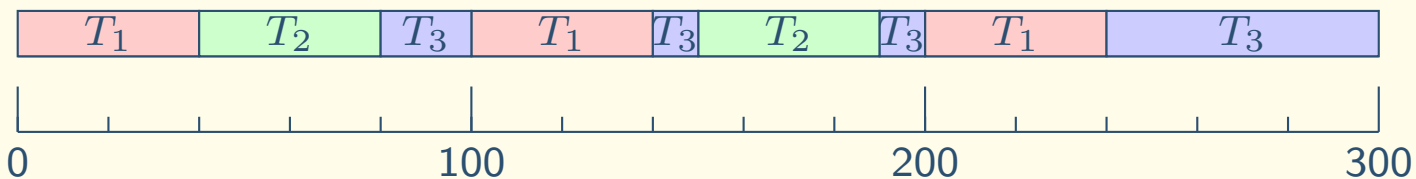
$$80 +$$

$$e_3 \leq$$

$$100 =$$

$$2 * P_2$$

$$300$$



Einführung

Annahmen

RMS

Rate Monotonic  
Scheduling

Beispiel

Schedulekriterium I

Schedulekriterium II

**Erweiterungen**

EDF

Probleme der  
besprochenen Ansätze

Task Assignment

Literatur

- "Deferred Server": Sporadische Tasks zulassen
- "Transient Overloads": kritische Tasks bevorzugen
- "Deadline Monotonic Scheduling": Deadline  $\neq$  Periode

Einführung

Annahmen

RMS

EDF

**Earliest Deadline First**

Beispiel

Schedulekriterium

Probleme der  
besprochenen Ansätze

Task Assignment

Literatur

- online Verfahren
- Task Prioritäten sind dynamisch
- Höchste Priorität hat der Task mit der nächsten Deadline
- Tasks müssen nicht periodisch sein
- optimaler Algorithmus bei dynamischen Prioritäten
- Falls ein gültiger Schedule existiert, so wird er mit EDF gefunden.

Einführung

Annahmen

RMS

EDF

Earliest Deadline First

**Beispiel**

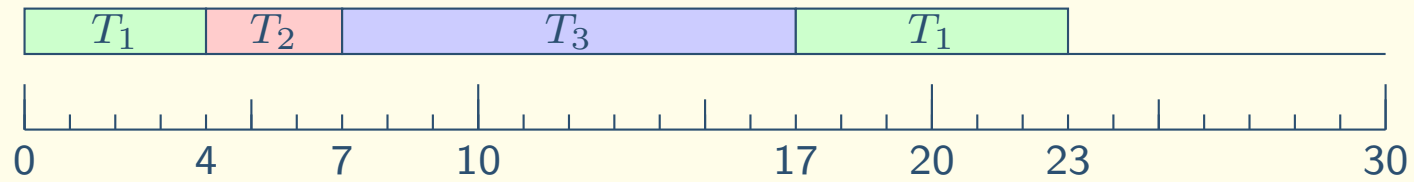
Schedulekriterium

Probleme der  
besprochenen Ansätze

Task Assignment

Literatur

Task	$I_i$	$e_i$	$D_i$
$T_1$	0	10	30
$T_2$	4	3	10
$T_3$	5	10	25



Einführung

Annahmen

RMS

EDF

Earliest Deadline First

Beispiel

**Schedulekriterium**

Probleme der  
besprochenen Ansätze

Task Assignment

Literatur

Unter der Annahme, dass es sich um ein periodisches Taskset handelt, bei der die relative Deadline gleich der Periode ist:

$$\sum_{i=1}^n e_i / P_i \leq 1$$

Einführung

Annahmen

RMS

EDF

Probleme der  
besprochenen Ansätze

**Probleme**

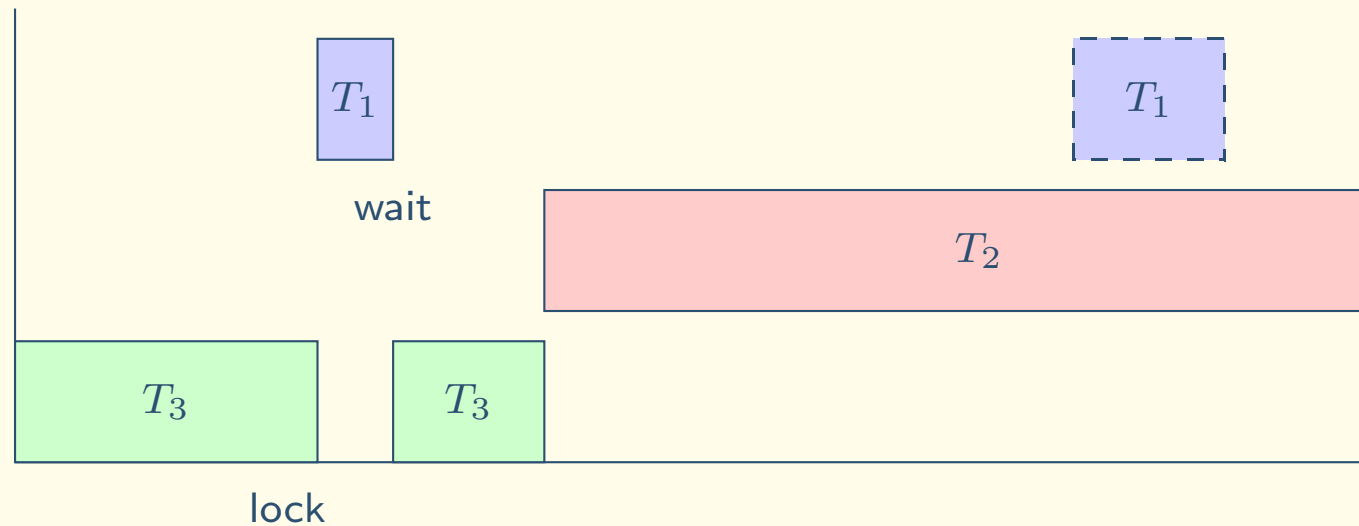
Priority Inheritance

Priority Ceiling

Task Assignment

Literatur

- Synchronisationen zwischen Tasks
  - Priority Inversion:  
Task mit kleinerer Priorität blockt Task mit hoher  
Priorität





Einführung

Annahmen

RMS

EDF

Probleme der  
besprochenen Ansätze

Probleme

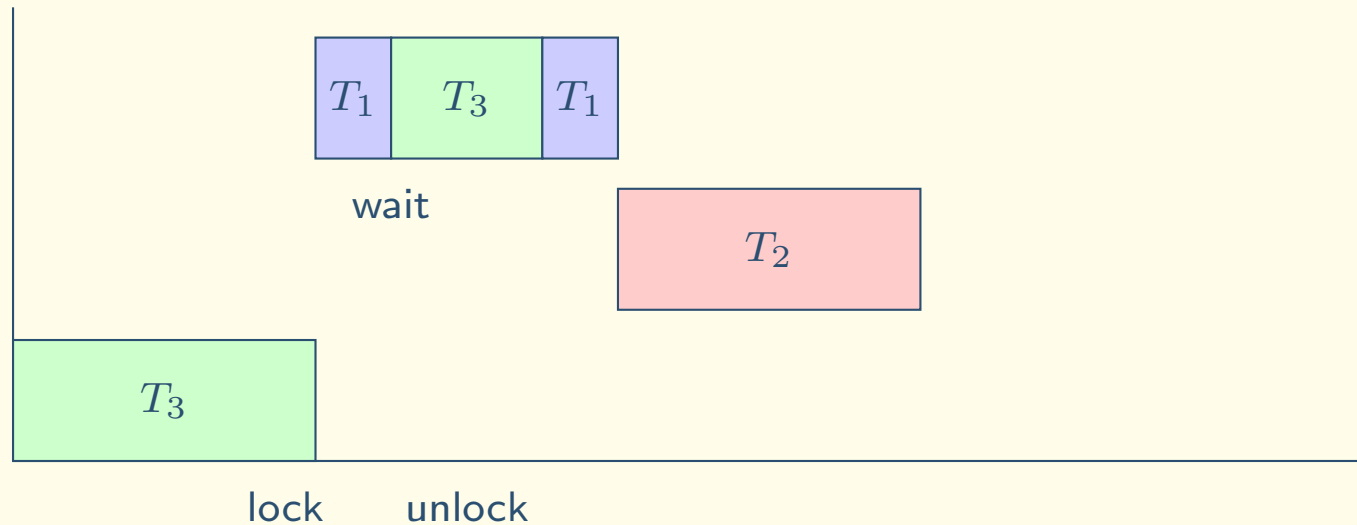
**Priority Inheritance**

Priority Ceiling

Task Assignment

Literatur

- Der niedrig priore Task erbt die Priorität des Tasks der auf die kritische Sektion wartet



- kann noch zu Deadlocks führen
- hebt im schlimmsten Falle die Priorität einer Task  $n - 1$  mal an

Einführung

Annahmen

RMS

EDF

Probleme der  
besprochenen Ansätze

Probleme

Priority Inheritance

Priority Ceiling

Task Assignment

Literatur

- Betreten der kritischen Sektion führt dazu, dass der Task die Priorität der höchsten Task bekommt, die ebenfalls diese Sektion betreten kann.
- verhindert Deadlocks

Critical Section	Accessed by	Priority Ceiling
$S_1$	$T_1, T_2$	$P(T_1)$
$S_2$	$T_1, T_2, T_3$	$P(T_1)$
$S_3$	$T_3$	$P(T_3)$
$S_4$	$T_2, T_3$	$P(T_2)$

Einführung

Annahmen

RMS

EDF

Probleme der  
besprochenen Ansätze

Task Assignment

**Taskzuweisung bei  
mehreren CPUs**

Taskzuweisungs-  
Algorithmen

I  
Taskzuweisungs-  
Algorithmen

II

Literatur

- NP-vollständig
- Heuristiken für die Zuweisung

Einführung

Annahmen

RMS

EDF

Probleme der  
besprochenen Ansätze

Task Assignment

Taskzuweisung bei  
mehreren CPUs

Taskzuweisungs-  
Algorithmen  
I

Taskzuweisungs-  
Algorithmen  
II

Literatur

- Utilization-Balancing
  - Last soll gleichmäßig aufgeteilt werden
  - Task nacheinander dem Prozessor mit der höchsten Idlezeit zuweisen
- Next-Fit für RMS
  - Tasks werden nach ihrer Last in Klassen eingeteilt
  - Anhand der Klassen Verteilung auf Prozessoren
- Bin-Packing
  - Taskzuweisung auf Standardproblem zurückführen:  
 $k$  Behälter mit Größe  $g$  für...

Einführung

Annahmen

RMS

EDF

Probleme der  
besprochenen Ansätze

Task Assignment

Taskzuweisung bei  
mehreren CPUs  
Taskzuweisungs-  
Algorithmen  
I

Taskzuweisungs-  
Algorithmen  
II

Literatur

- Myopic Offline Scheduling
  - im Prinzip Backtracking mit speziellem Gültigkeitstest
  - berücksichtigt auch nicht unterbrechbare Tasks
- Focused Addressing and Bidding
  - Unterteilung in kritische und unkritische Tasks
  - kritische Tasks werden statisch verteilt
  - unkritische Tasks landen zuerst bei irgendeinem Prozessor
  - Lastverteilung wenn Schedule nicht gültig
- Buddy Strategy
  - ähnlich wie FAB, Unterscheidung in Wahl der Offload-CPU
  - Einordnung in 3 Kategorien: underloaded, fully loaded, overloaded

Einführung

Annahmen

RMS

EDF

Probleme der  
besprochenen Ansätze

Task Assignment

Literatur

**Literatur**

- C. M. Krishna and K. G. Shin, Real-Time Systems, McGraw-Hill, 1997.
- Jane W. S. Liu, Real-Time Systems, Prentice Hall, 2000.