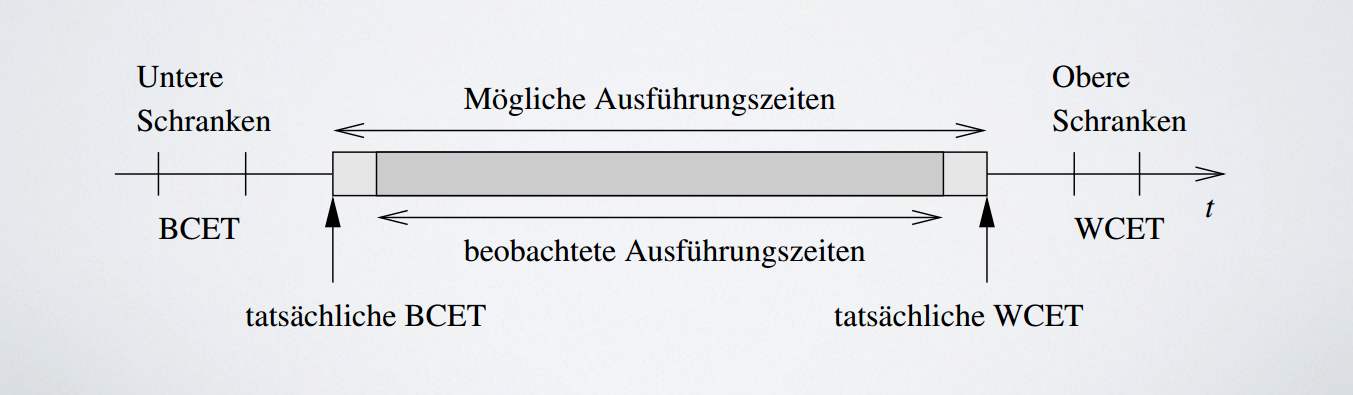
**Scheduling**

* Zuordnung von Tasks zu Ressourcen
* Meistens in Systemen mit Echtzeit-Anforderungen

Es muss immer abgeschätzt werden welches Scheduling-Verfahren anhand der längsten Ausführungszeit Sinnvoll ist.

1. Worst Case Execution Time (WCET)
   1. Längst mögliche Ausführungszeit
   2. Obere Schranke für Scheduling Verfahren
2. Best Case Execution Time (BCET)
   1. Kürzest mögliche Ausführungszeit
   2. Untere Schranke für Scheduling Verfahren



Scheduringalgorithmen werden unterschieden nach

1. Harten Deadlines

Verfehlen einer Zeitbedingung ist ein Fehler!

* 1. Periodisch/aperiodisch
  2. Präemptiv/nicht präemptiv
  3. Statisch/dyamisch

1. Weichen Deadlines

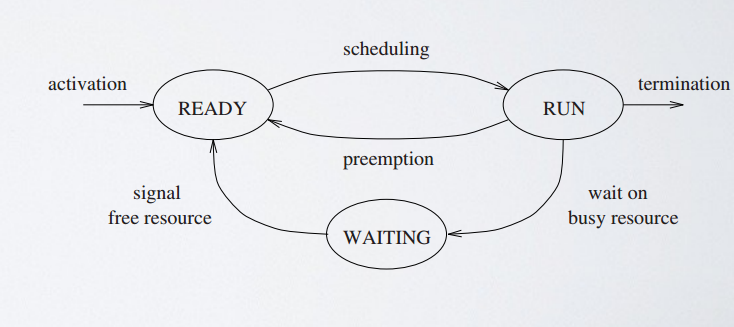
Verfehlen einer Zeitbedingung ist kein Fehler, statistisch sollte die Zeitbedingung eingehalten werden!



**Zustände eines Tasks**

1. Ready, bereit zur Ausführung
2. Waiting/Blocked,  
    blockiert durch nicht verfügbare Ressource
3. Run, in Ausführung

Task können für wichtigere Tasks unterbrochen werden



Tasks sind

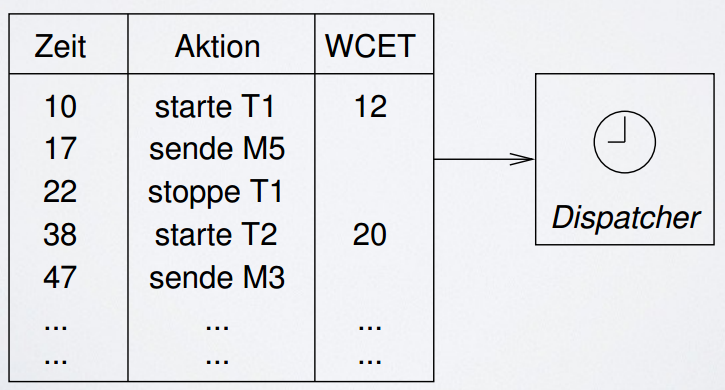
1. Periodisch, wenn sie alle p Zeiteinheiten ausgführt werden
2. Aperiodisch, alternativ

Scheduler sind

1. Nicht-präemptiv, wenn sie die Ausführung eines Tasks abwarten
2. Präemptiv, wenn sie die Ausführung eines Tasks unterbrechen
3. Dynamisch, wenn die Entscheidung zur Laufzeit getroffen wird(auch on line Scheduler genannt)
4. Statisch, wenn die Entscheidung zur Entwurfszeit (vor Laufzeit) festgelegt werden(off-line Scheduler).

**Statischer Scheduler**

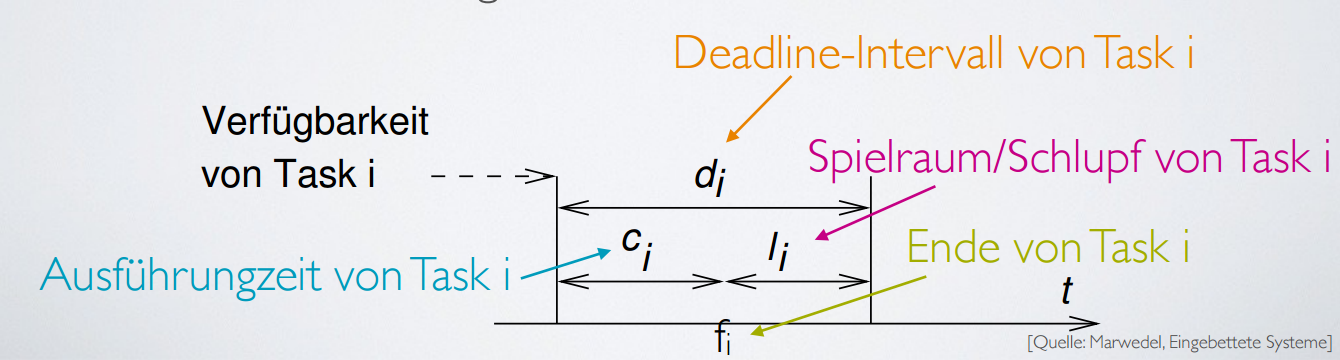
1. Startzeiten werden in Tabellen abgelegt
2. Ein Dispatcher startet Tasks zu angegebener Zeit
3. Ablaufplanung erfolgt anhand der Priorität



Bei abhängigen Tasks, Nutzung eines Tasks-Graphen

**Aperiodisches Scheduling**

1. Maximale Verspätung
   1. Maximale Differenz zwischen Ausführungsende und Deadline
2. Deadline-Intervall:
   1. Zeit zwischen Verfügbarkeit und Deadline



**Round Robin mit Zeitschlitzen**

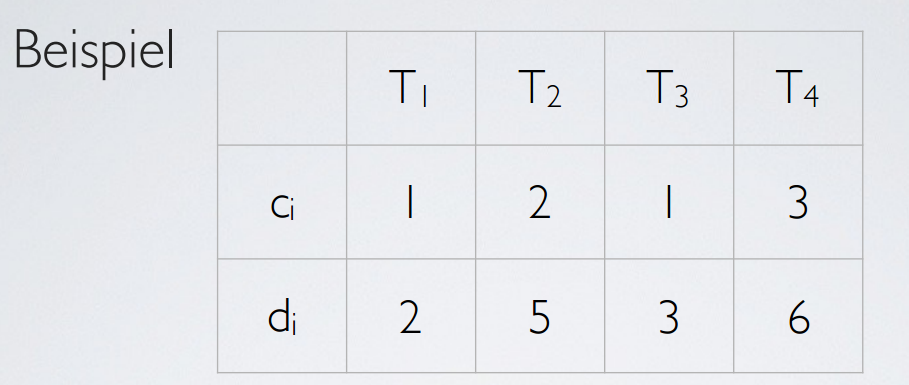
1. Sequentielle Abarbeitung der Tasks(nach FIFO)
   1. Einreihung in Queue(Waiting)
   2. Erster Task aus Queue zur Ausführung(Running)

Dauer der Ausführung : eine Zeitscheibe(Timesclice/Quantum)

Keine Unterbrechungen!

**Erliest Due Date**

1. Abarbeitung der Taks anhand der Deadlines (frühste Deadline first)
2. Sehr Komplex zu Sortieren O(nlog(n))



Lösung: T1 T3 T2 T4(Tastende: f1=1,f3=2,f2=4,f4=7)

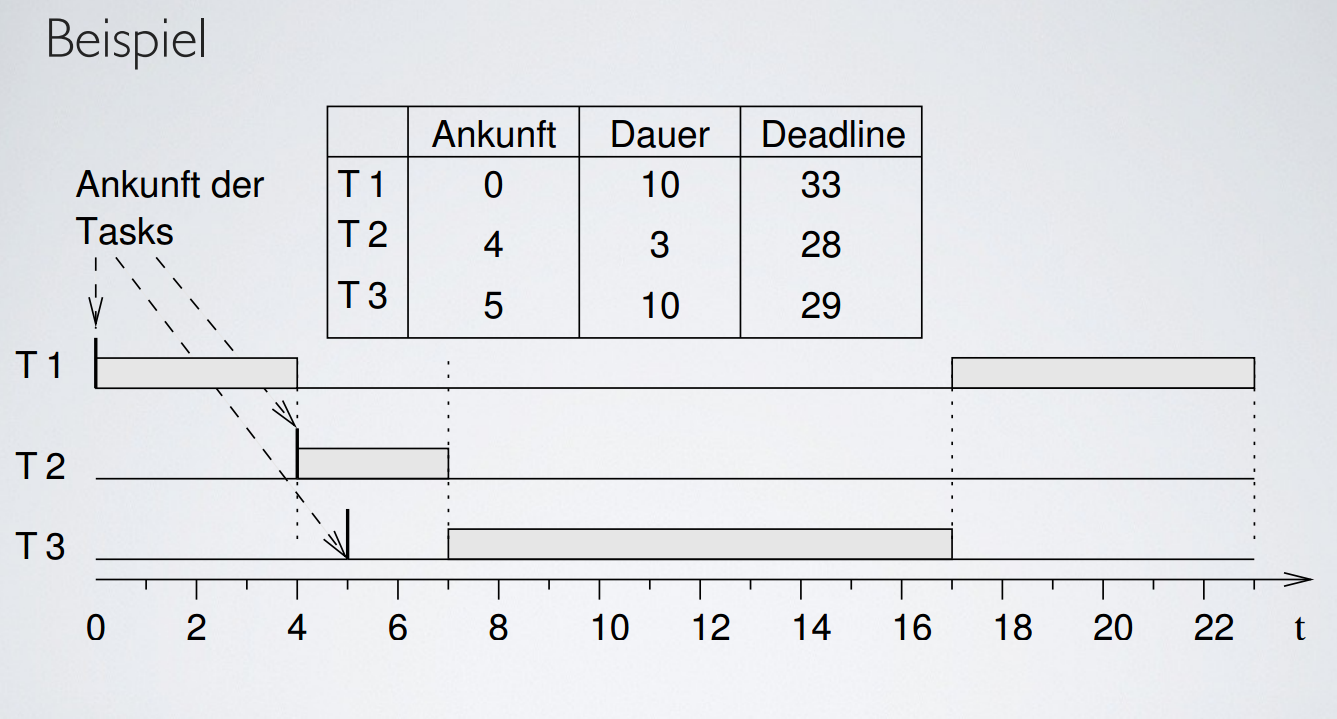
Max Verspätung : 1 (T4)

**Erliest Deadline First EDF**

Zu jedem Zeitpunkt wird Task mit der frühesten absoluten Deadline ausgeführt, ggf. Unterbrechen von Tasks

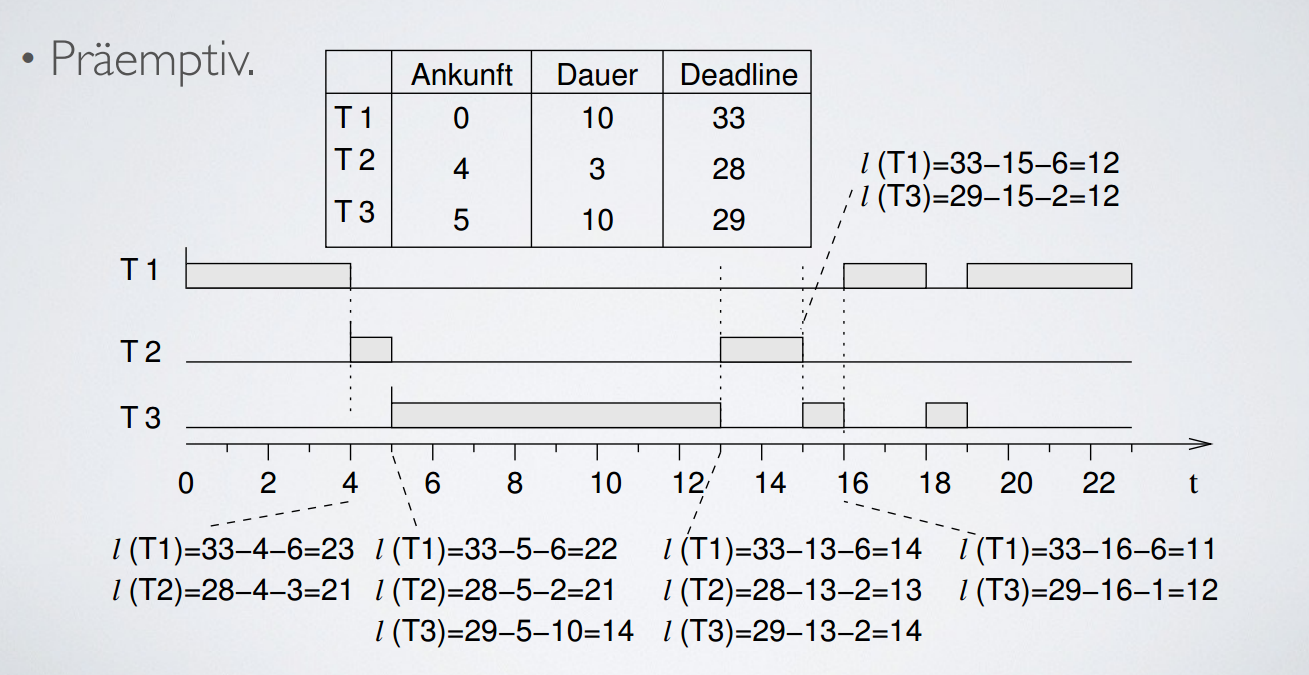
Optimal für Minimierung der maximalen Verspätung

Warteschlange sortiert nach Deadlines



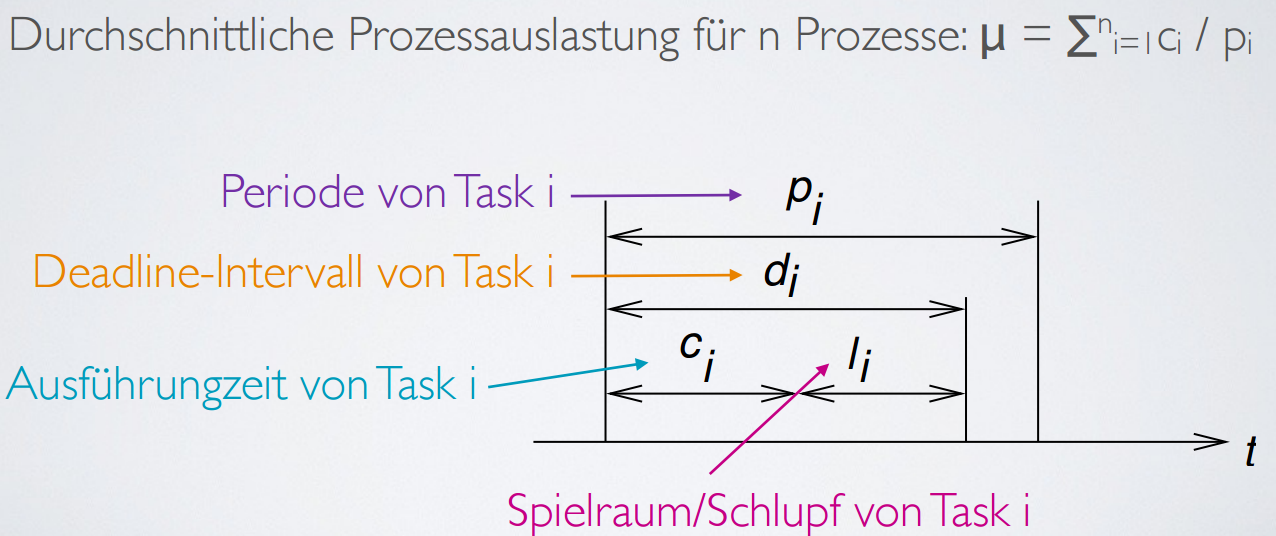
**Least-Laxity (Geringster Schlupf)**

Prioritäten der Tasks sind monoton fallende Funktion ihres Schlupfs. Schlupf verändert sich dynamisch.



**Periodisches Scheduling**

Jede Ausführung eines Tasks heißt Job -> Jobs eines Tasks immer gleich lang

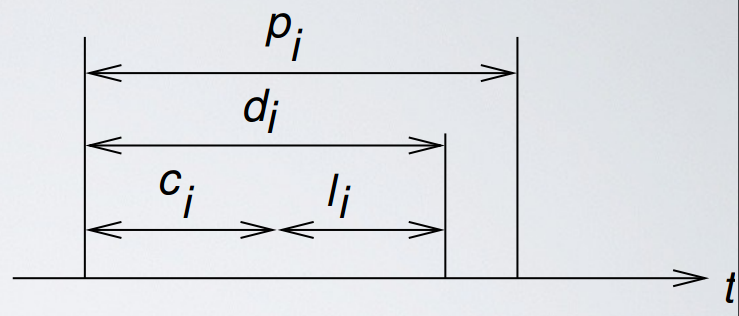


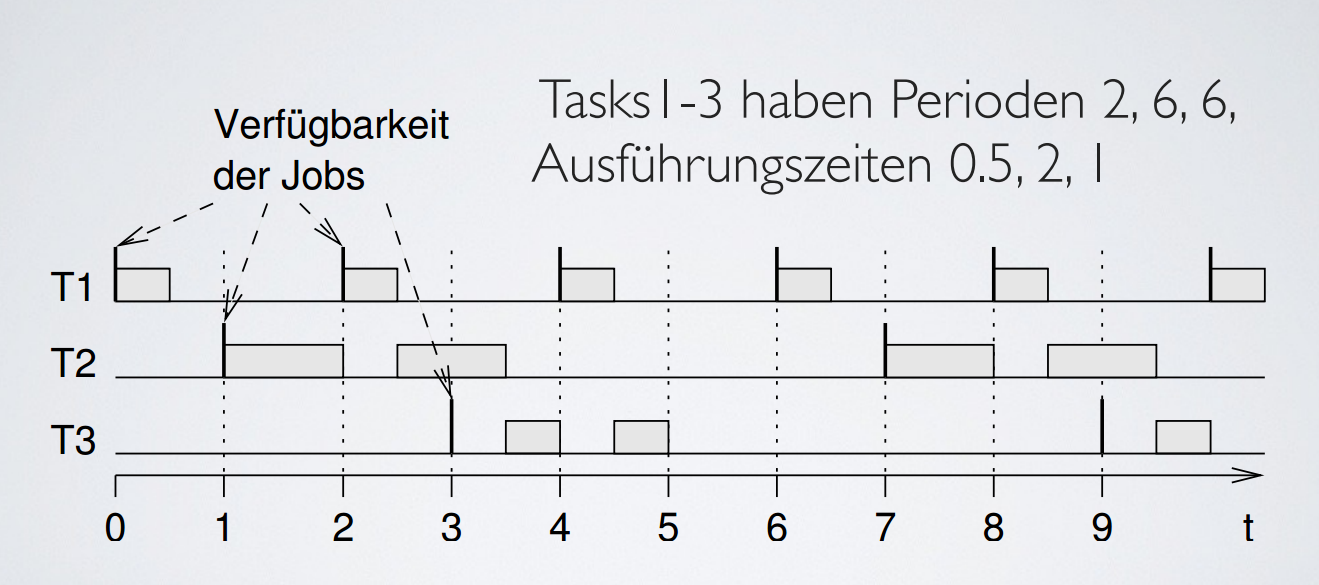
**Rate Monotonic Scheduling**

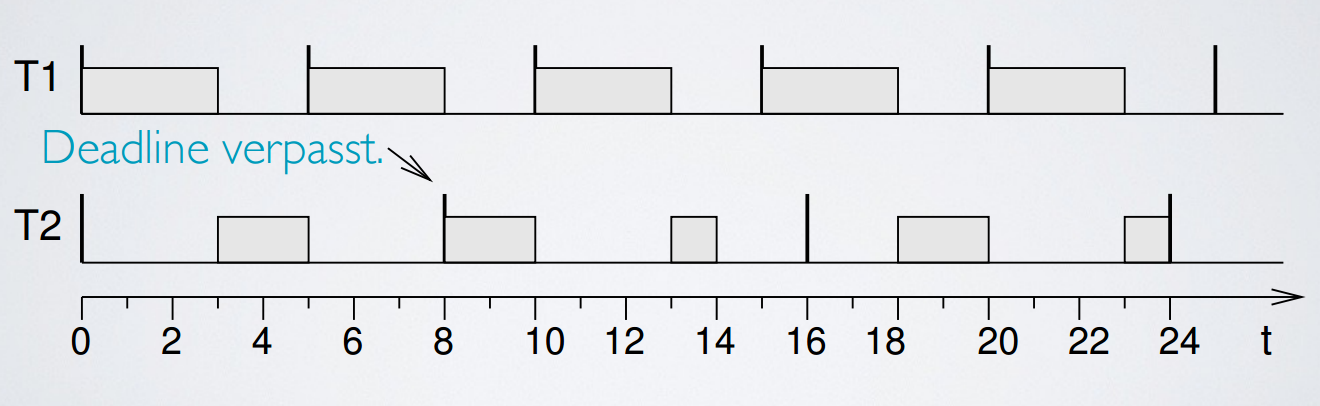
Voraussetzung: unabhängige Tasks, di = pi, ci konstant und bekannt

bei n Tasks Einhaltung von Deadlines für Auslastung μ = ∑n i=1ci / pi ≤ n(21/n-1)

Priorität ist monoton fallende Funktion ihrer Periode, also haben Tasks mit kurzer Periode hohe Priorität







**Rate Monotonic Analysis / RMA**

Analyse des Rate Monotonic Schedulings

Betrachtung des Falls mit der schlechtesten Antwortzeit