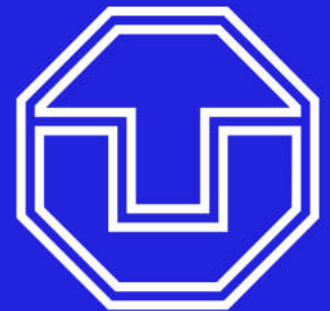


Quantitative Methoden

Betriebssysteme

Claude-J. Hamann
TU Dresden



Problem und Gegenstand

- **Problem**

Erfüllen von QoS-Anforderungen mit zeit- bzw. größenbeschränkten Ressourcen

- **Gegenstand**

-  Scheduling

basierend auf

- deterministischen Modellen
- probabilistischen Modellen

Hauptspeicherverwaltung

Externspeicherzugriff

Dateisysteme

-  Echtzeitsysteme

Literatur

WECK, G.: Prinzipien und Realisierung von Betriebssystemen.
B. G. Teubner 1983.

- LIU, J. W. S.: Real-Time Systems. Prentice-Hall, 2000

STANKOWIC, J. A., et al.: Implications of Classical Scheduling
Results for Real-Time Systems. In: Computer 6/1995.

DOWDY, L.; C. LOWERY: P.S. to Operating Systems.
Prentice-Hall, 1993.

PFLUG, G.: Stochastische Modelle in der Informatik.
B. G. Teubner 1986.

Scheduling – Einführung

- **Begriff**

Vorgehensweise zur **Einplanung** von Aufgaben, die durch ein aktives Betriebsmittel zu bearbeiten sind

Entscheidungsstrategien, die die Reihenfolge festlegen, in der sich Prozesse um den Prozessor (allgemeiner: um ein Betriebsmittel) bewerben müssen bzw. in der sie aus einer Warteschlange (für das Betriebsmittel) ausgewählt werden

- **Aufgabe der Schedulingtheorie**

Entwicklung und Bewertung (!) derartiger Strategien

Scheduling – Einführung

- **Ziele**

hohe Prozessorauslastung

größtmöglicher Durchsatz

minimale Gesamtbearbeitungszeit

$$\left. \begin{array}{l} \eta/U \\ D \\ t_g \end{array} \right\} \approx$$

geringe durchschnittliche Verweilzeit

minimale Antwortzeit

garantierte Reaktionszeit

$$\left. \begin{array}{l} \bar{t}_v \end{array} \right\} \approx$$

Gerechtigkeit

- **Einordnung und Abgrenzung**

Ablaufplanung (Teilgebiet der Operationsforschung)

Prozeßauswahl (System-S.) – Prozessorzuteilung (Dispatching)

Strategie – Algorithmus – Implementation

Scheduling – Einführung

- **Ablaufplan (Schedule)**

zeitabhängige Zuordnung von Prozessen zu Prozessoren

oft: graphische Darstellung der Prozessorzuteilung in Form eines GANTT-Diagramms

- **Prozeß - Thread - Job - Task - Auftrag - Vorgang - ...**

- **Klassifikationsgesichtspunkte**

Ein- / Mehrprozessorsysteme

Bearbeitung ohne / mit Prozessorentzug

Deterministische / probabilistische Modelle

Echtzeitbedingungen

Scheduling – Deterministische Modelle

▪ Modellannahmen

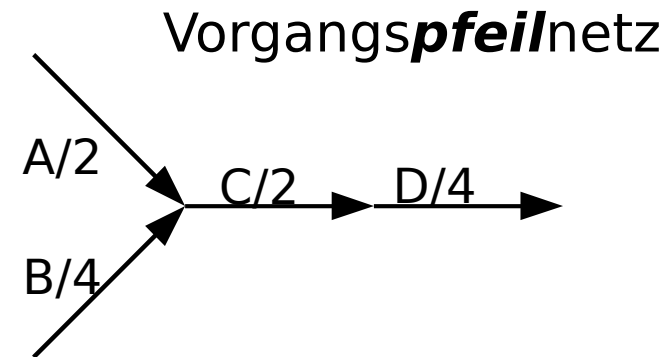
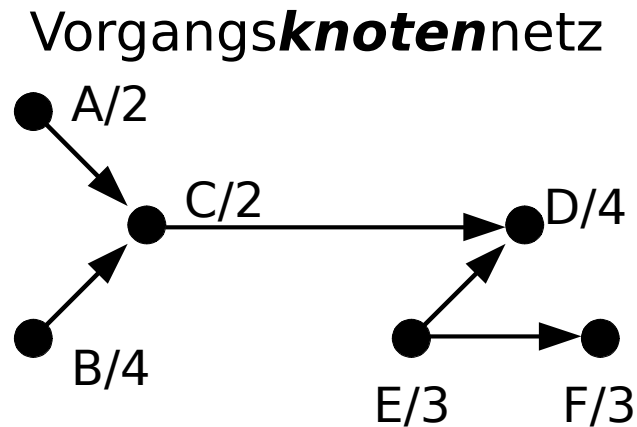
Gegeben:

$J = \{J_1, \dots, J_n\}$	Menge von Jobs
$R \subseteq J \times J$	Präzedenzrelation
$t: J \rightarrow \mathbb{R}^+$	Abbildung, wobei $t(J_i) =: t_i$

- durch Messung oder Rechnung ermittelte tatsächliche (konstante) Ausführungszeit
 - auf Erfahrung beruhende mittlere Zeit
 - abgeschätzte maximal mögliche Ausführungszeit (WCET)
-
- Anwendungsbereich: (annähernd) konstantes Aufgabenprofil

Scheduling - Deterministische Modelle

- Graphische Darstellung

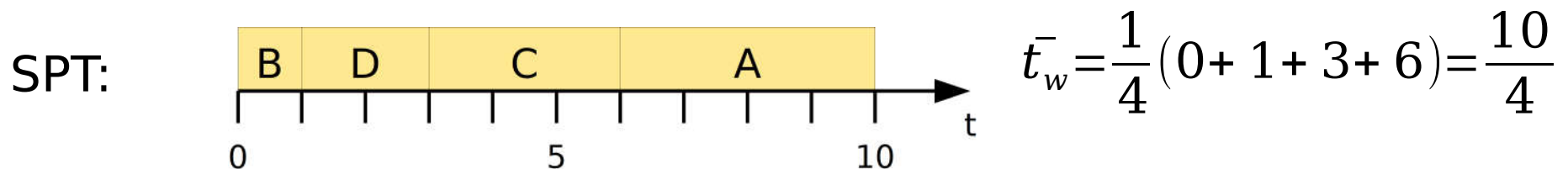
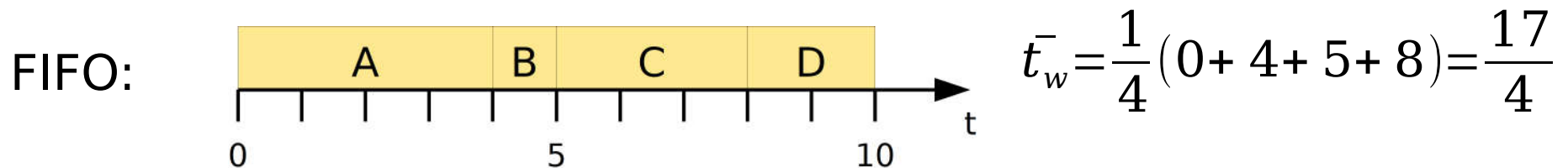


Scheduling in 1-Prozessor-Systemen ohne Entzug

- **FIFO / LIFO**
- **SPT „Shortest Processing Time“**
ist bei $R = \emptyset$ optimal bzgl. $\bar{t}_v \rightarrow \text{Min!}$

Beispiel:

J_i	A	B	C	D
t_i	4	1	3	2



Bei $R \neq \emptyset$ ist das Problem NP-vollständig!

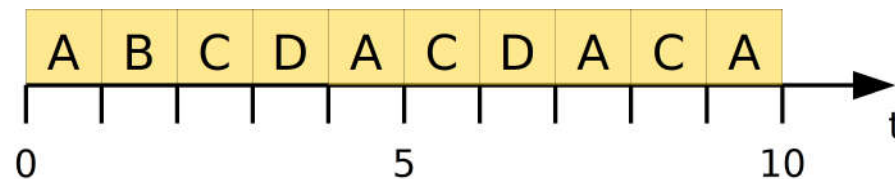
Scheduling in 1-Prozessor-Systemen mit Entzug

- **RR „Round Robin“**

Prozeßwechsel mit konstantem Zeitquant Q

Beispiel:

J_i	A	B	C	D	
t_i	4	1	3	2	$Q = 1$



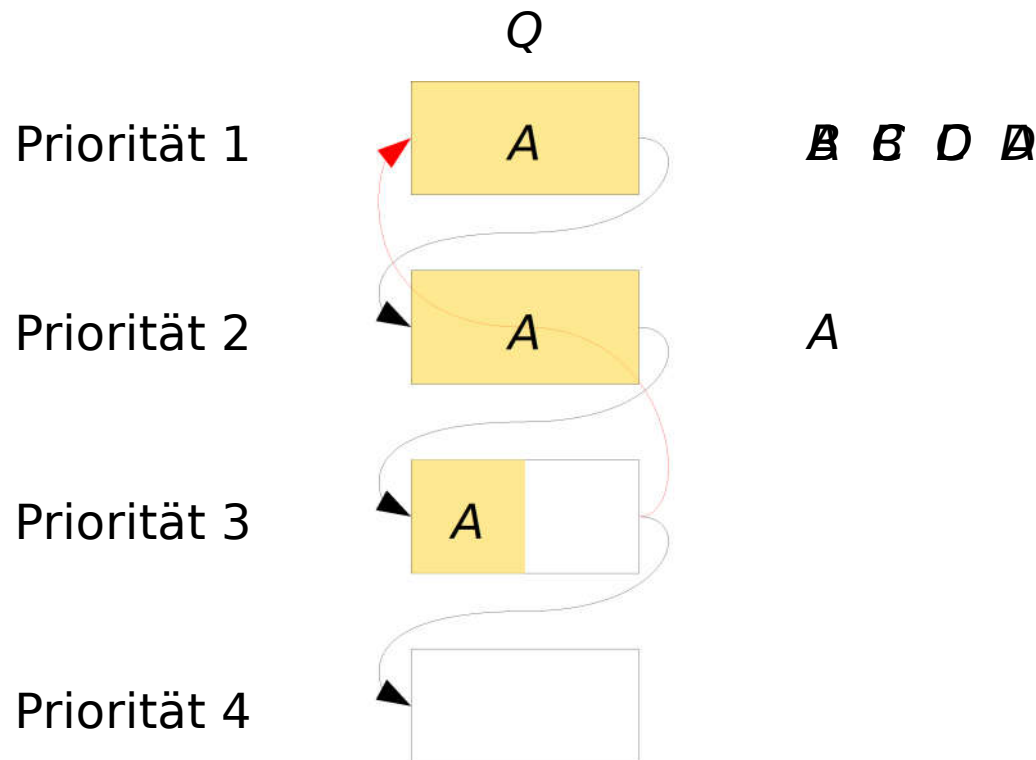
$$\bar{t}_w = \frac{1}{4} (6 + 1 + 6 + 5) = \frac{18}{4}$$

Problem: Größe von Q

Scheduling in 1-Prozessor-Systemen mit Entzug

- **MLF „Multilevel-Feedback“**

Prozeßwechsel mit Zeitquant Q und Prioritäten



Scheduling in 1-Prozessor-Systemen

- **Rechenintensive vs. E/A-intensive Jobs**

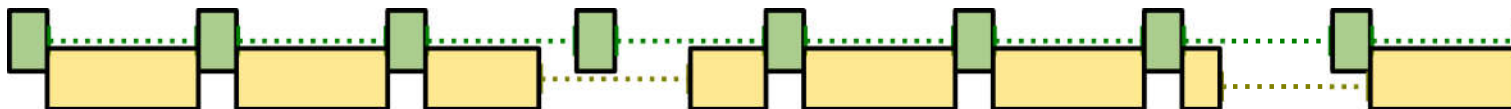
Antwortzeit und Durchsatz abhängig von Priorität



Priorität(A) > Priorität(B)



Priorität(B) > Priorität(A)



Scheduling in Mehrprozessor-Systemen

m identische Prozessoren. Enumeration: Aufwand $O(e^{\text{jobanzahl}})$

- **Optimalitätskriterium $t_g \rightarrow \text{Min!}$**

R bel.: polynomialer Algorithmus nur für $m = 2$, $t_i = \text{const.}$ bekannt

$R = \emptyset$: $m = 1$ trivial

$m > 1$: Approximation

LPT „Largest Processing Time“

- **Optimalitätskriterium $\bar{t}_v \rightarrow \text{Min!}$**

$R = \emptyset$: SPT ist optimal (sonst NP-vollständig)

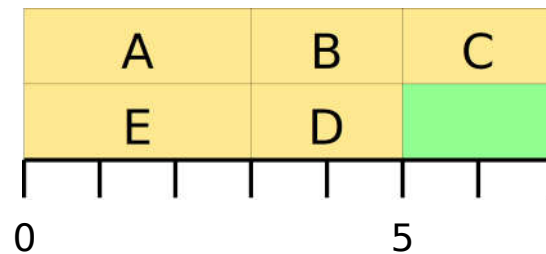
Scheduling in Mehrprozessor-Systemen

- Beispiele

$m = 2$

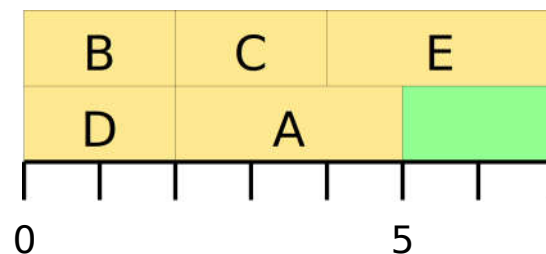
J_i	A	B	C	D	E
t_i	3	2	2	2	3

LPT:



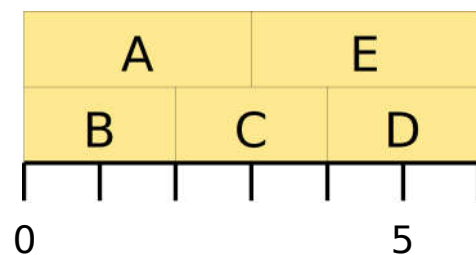
$$t_g = 7 \quad \bar{t}_w = \frac{11}{5}$$

SPT:



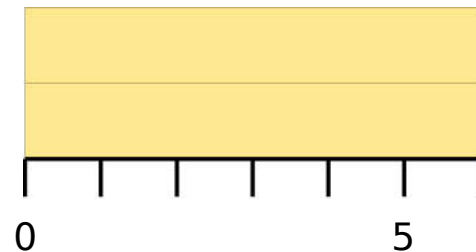
$$t_g = 7 \quad \bar{t}_w = \frac{8}{5}$$

Opt. bzgl.
 t_g und \bar{t}_w



$$t_g = 6 \quad \bar{t}_w = \frac{9}{5}$$

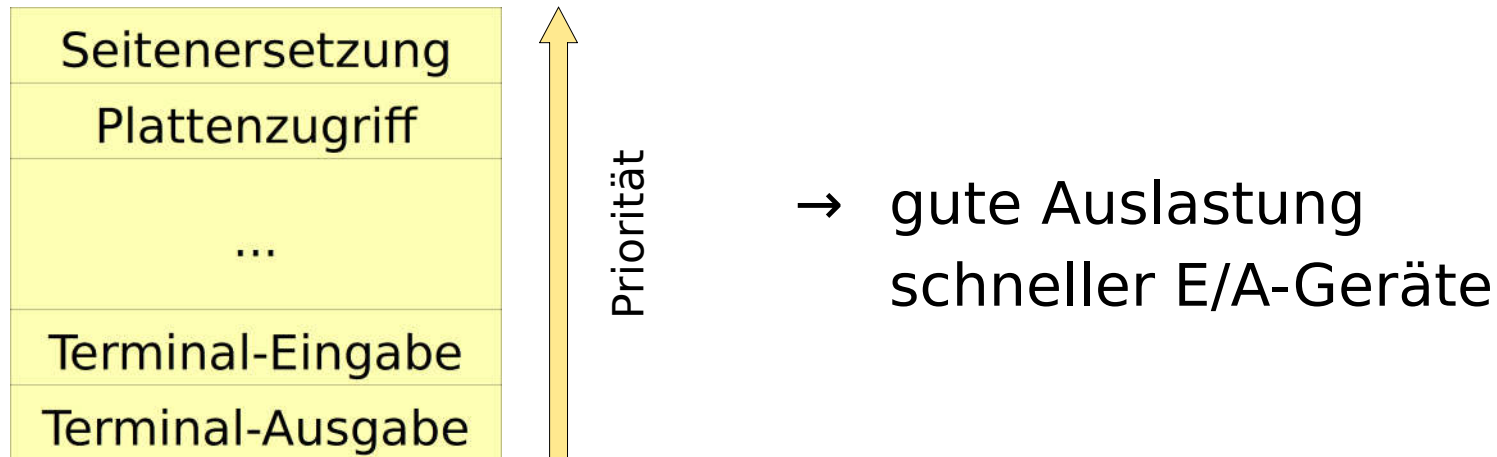
Entzug:



$$t_g = 6 \quad \bar{t}_w = \frac{8}{5}$$

Fallstudie: Unix (konzeptionell)

▪ Statische Prioritäten im Kern-Modus



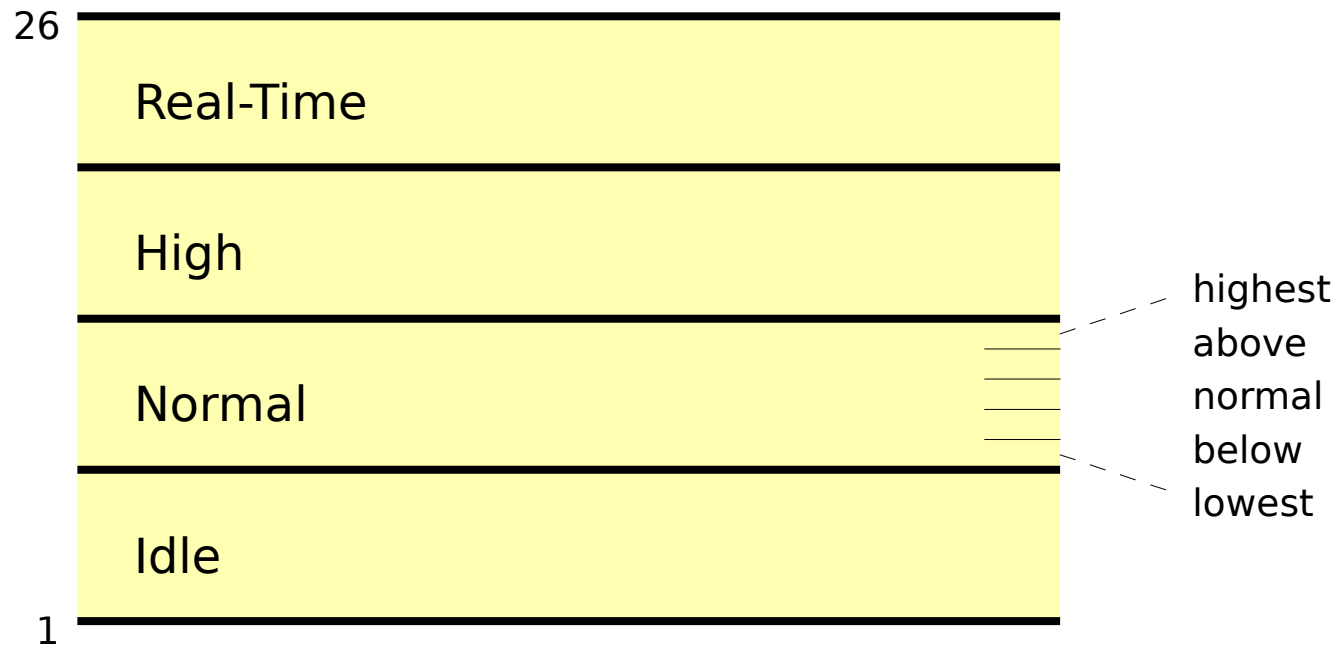
▪ Dynamische Prioritäten im Nutzer-Modus

Priorität = $f(\text{Basis-Priorität, nice-Wert, CPU-Nutzung})$

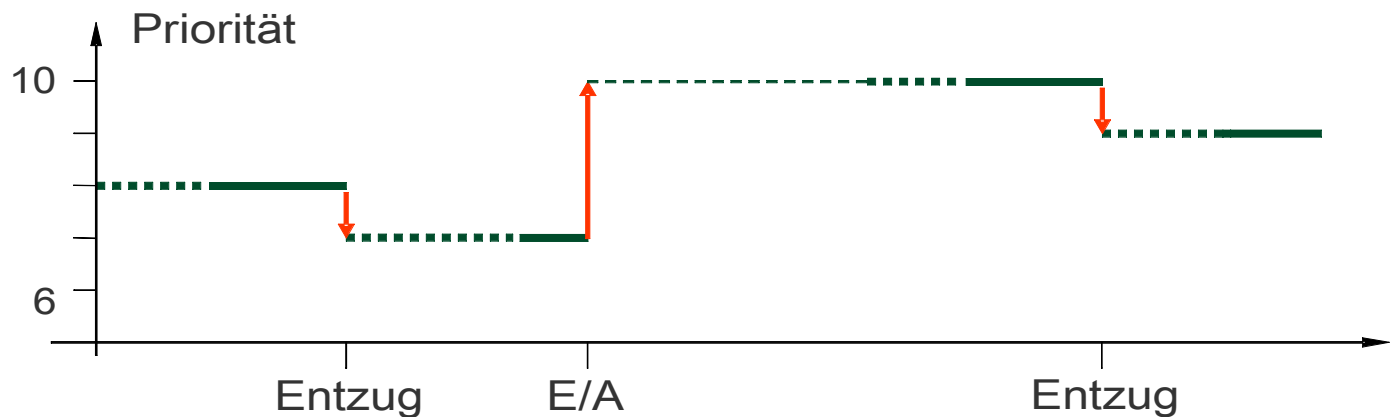
→ Bevorzugung interaktiver Anwendungen

Fallstudie: Windows

- Vier „Prioritätsbänder“ mit jeweils fünf Stufen



- Dynamische Priorität entsprechend E/A-Aktivität



Echtzeit-Scheduling – Grundbegriffe

- **Job**

Planungseinheit für Scheduling

- e Ausführungszeit, Bearbeitungszeit (execution time)
- r Freigabezeit, Bereitzeit (release time)
- d Zeitschranke, Frist (deadline)

- **Task**

Menge „zusammengehörender“ Jobs

speziell: ***Jobnetz*** oder ***periodische Task***

- **Deadline**

hart / weich

Echtzeit-Scheduling – Grundbegriffe

- **Schedule (Ablaufplan)**

zeitliche Zuordnung von Jobs zu Prozessoren

gültig (valid): Zuordnung verletzt keine der gegebenen Bedingungen

ausführbar (feasible): alle Zeitschranken werden eingehalten

- **Scheduling**

- **Einplanung:** Vorgehen (Algorithmus), das bei gegebener Taskbeschreibung einen Ablaufplan bestimmt
- **Prozessor-Zuordnung:** Auswahl eines Jobs durch Scheduler des Systems

Echtzeit-Scheduling – Grundbegriffe

- **Einplanbarkeit**

Taskmenge ist **einplanbar** (*schedulable, feasible*) bei einem Scheduling-Algorithmus, wenn der Algorithmus einen ausführbaren Ablaufplan erzeugt

- **Admission (Zulassung)** 

Verfahren, das die Einplanbarkeit einer Taskmenge entscheidet

- **Optimalität (bzgl. Einplanbarkeit)**

eines Scheduling-Verfahrens in einer Klasse **C** von Verfahren:
erzeugt für jede Taskmenge **T** einen ausführbaren Ablaufplan, sofern **T** überhaupt mit einem Verfahren aus **C** eingeplant werden kann

Echtzeit-Scheduling – Modellannahmen

▪ **Deterministisches Modell**

jede Task T_i ist periodische Folge von Jobs, Periode p_i

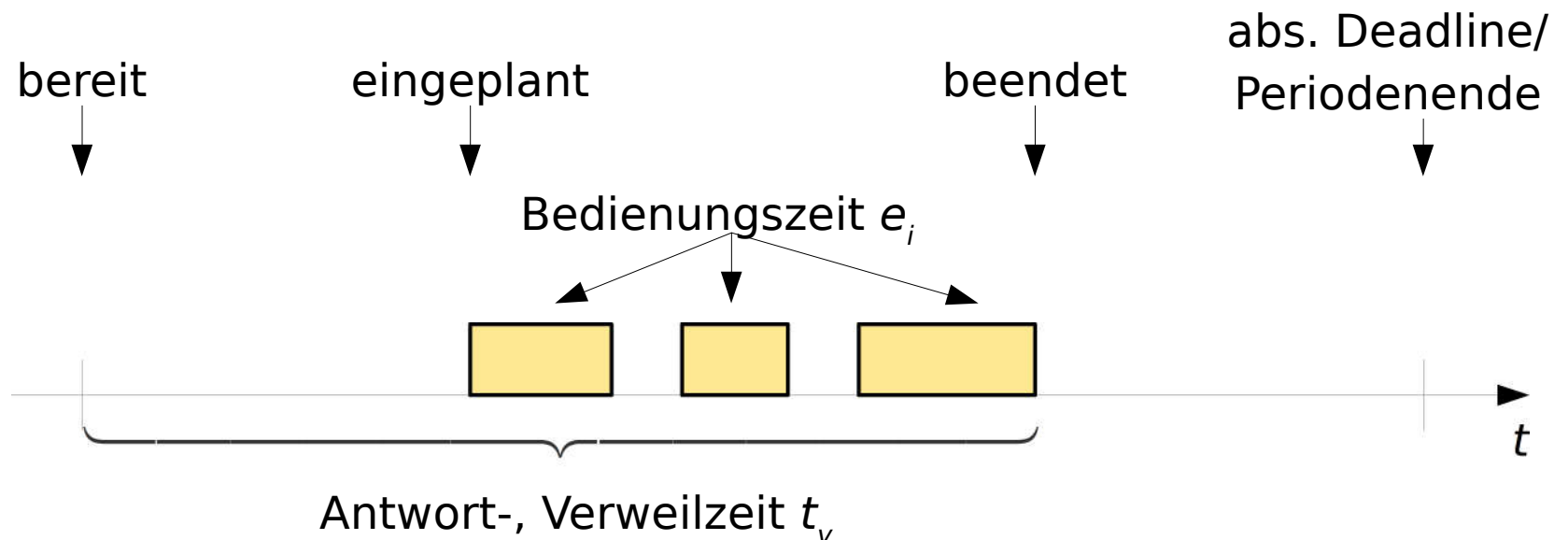
Periode ist zugleich Zeitschranke (relative Deadline d_i)

Bearbeitungszeit e_i ist konstant

Prozessor ist entziehbar

Tasks sind voneinander unabhängig („in Zeit und Raum“)

System-Scheduling prioritätsbasiert



Echtzeit-Scheduling – Verfahren

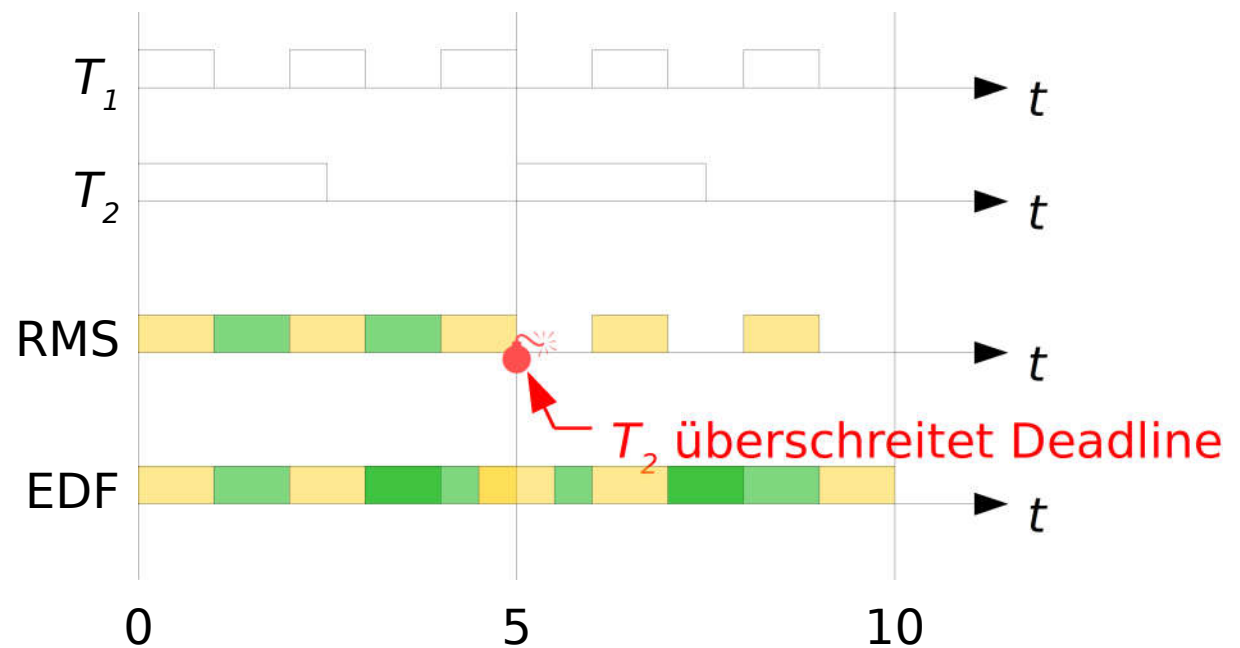
- **RMS „Rate Monotonic Scheduling“** LIU/LAYLAND, 1973
statische Taskpriorität proportional zu Ankunftsrate

- **EDF „Earliest Deadline First“**

Task mit nächstgelegener Deadline hat aktuell höchste Priorität

- **Beispiel**

T_1 hat höhere
Priorität als T_2



Echtzeit-Scheduling – Eigenschaften

- **Optimalitätseigenschaft**

RMS ist optimal in der Klasse aller statischen, EDF in der Klasse aller (dynamischen) Scheduling-Algorithmen (in Einprozessor-Systemen für unabhängige, verdrängbare Tasks)

- **Existenz eines Ablaufplans (Admission-Kriterium)**

bei RMS, falls $\eta = \sum_{i=1}^n \frac{e_i}{p_i} \leq n \cdot (\sqrt[n]{2} - 1) \approx 0,83 \dots 0,69$ (ln 2)

bei EDF genau dann, wenn $\eta = \sum_{i=1}^n \frac{e_i}{p_i} \leq 1$

Scheduling - Probabilistische Modelle

- **Modellannahmen**

In zufälligem Ankunftsabstand T_a treffen Anforderungen an den Scheduler mit zufälligem Bedarf an Bedienungszeit T_b ein.

Anwendungsbereich: Interaktiver Betrieb, verteilte Systeme

- **Aufgabe**

Bestimmen von Bewertungsgrößen wie

T_w Wartezeit $T_v = T_w + T_b$ Verweilzeit

N_w Warteschlangenlänge

N_v Anzahl der im System befindlichen Forderungen

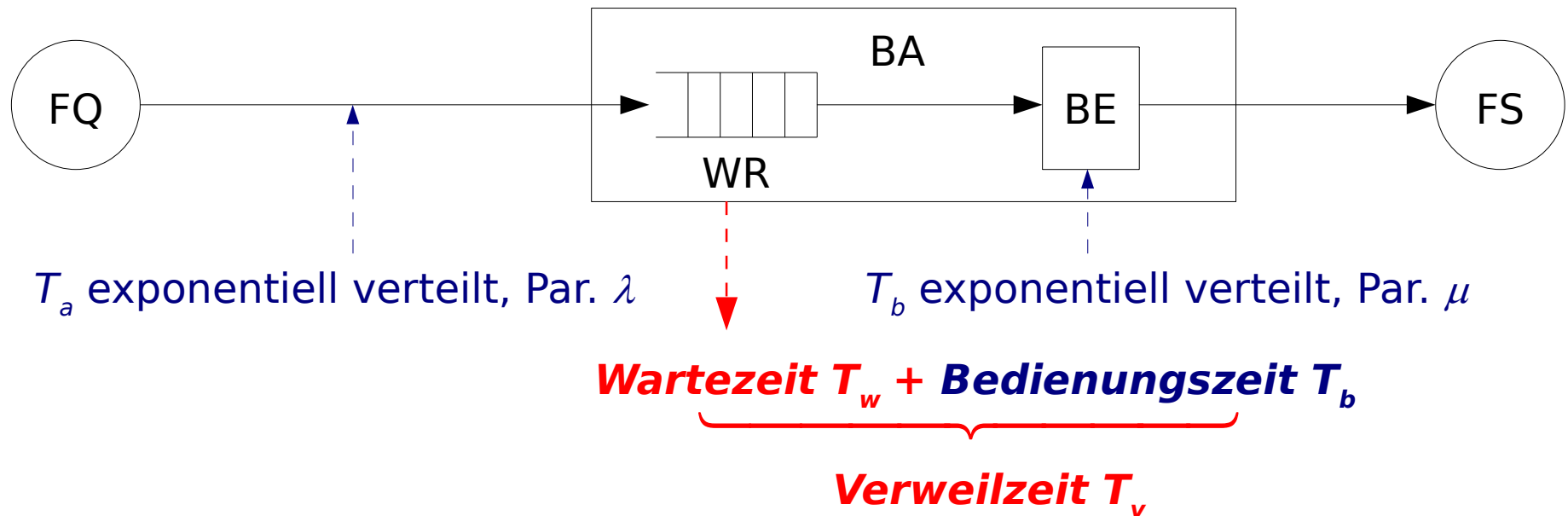
- **Mathematisches Instrumentarium**

Bedienungstheorie (Warteschlangentheorie)

Probabilistische Modelle – M/M/1/∞-System

▪ Struktur

BA: Bedienungsanlage FQ: Forderungsquelle
WR: Warteraum FS: Forderungssenke
BE: Bedieneinrichtung



▪ Bewertungsgrößen

$$E(T_w) = \frac{\rho}{\mu(1-\rho)}$$

$$E(T_v) = E(T_w) + E(T_b) = \frac{1}{\mu(1-\rho)}$$

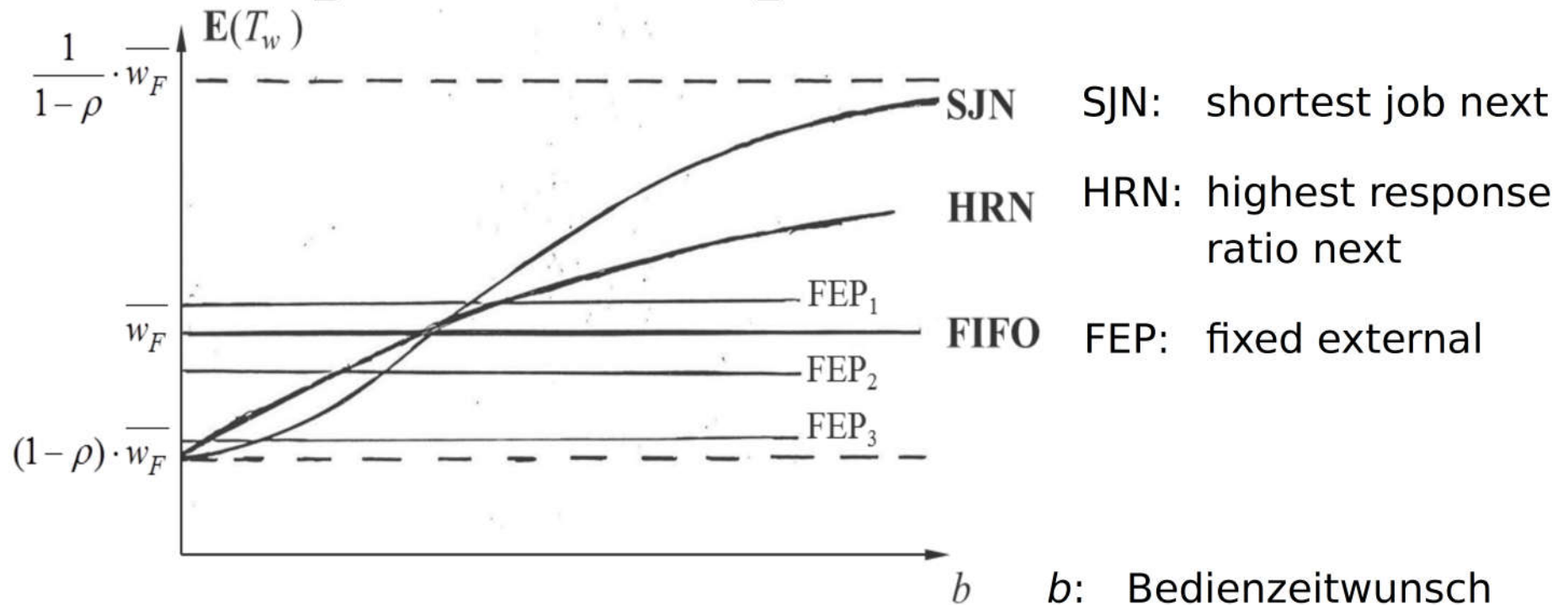
$$E(N_w) = \frac{\rho^2}{1-\rho}$$

$$E(N_v) = E(N_w) + E(N_b) = \frac{\rho}{1-\rho}$$

$$\rho = \frac{\lambda}{\mu} = \frac{E(T_b)}{E(T_a)}$$

Probabilistische Modelle – Anwendung

▪ Bewertung von Scheduling-Verfahren



FIFO: einfach. Gleichbehandlung aller Aufträge

SJN: Bevorzugung kürzerer Aufträge;
 \bar{t}_w wird bei $R = \emptyset$ minimal, falls alle b bekannt

HRN: größere Gerechtigkeit

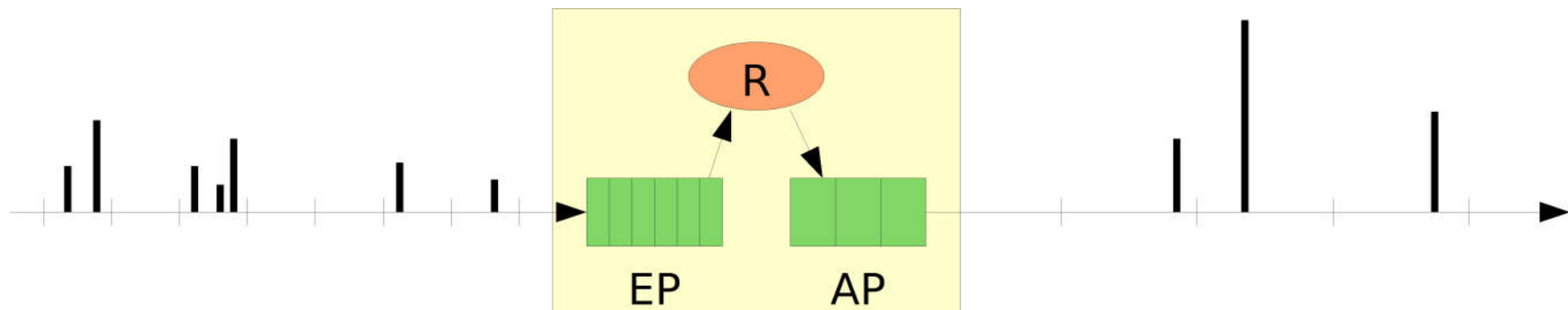
FEP: statisch. Prioritätszuordnung?

Ausblick: Quantitative Methoden

- **Bewertung von Speicher-Zuteilungsstrategien**
 - **Segmentierter Speicher**

Freispeicherlisten → externe Fragmentierung
Auffüllen und Kompaktifizieren → Aufwand
 - **Seitenorientierte Systeme**

Arbeitsmengen → Fenstergröße; Seitenflattern
statische Ersetzungsstrategien → Seitenfehlerrate
- !▪ **Pufferdimensionierung**



Ausblick: Quantitative Methoden

- **Externspeicher-Zugriff**

- **Scheduling**

Leistungsanalyse mittels Bedienungsmodellen/-netzen
Nutzung weiterer, nicht entziehbarer Ressourcen
(Prioritätsinversion, Berechnung von Blockierungszeiten)

Scheduling-Verfahren für flexible Applikationen

***Scheduling-Verfahren für nicht-periodische Tasks
und Tasks ohne Echtzeit-Anforderungen***

Admission-Algorithmen

- **Leistungsanalyse von Betriebssystem-Komponenten**



„Scheduling-Theorie“

Quantitative Methoden

1. Problem, Gegenstand und Grundbegriffe	2
2. Scheduling - Deterministische Modelle	7
2.1. Modellannahmen und -beschreibung	7
2.2. Scheduling in 1-Prozessor-Systemen	9
2.3. Scheduling in Mehrprozessor-Systemen	13
2.4. Fallstudien	15
3. Scheduling in Echtzeitsystemen	17
3.1. Grundlagen	17
3.2. Modellannahmen und Verfahren	20
4. Scheduling - Probabilistische Modelle	23
5. Ausblick	26