

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



Echtzeitsysteme

Literatur

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

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

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

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

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

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

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

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

- **Planungseinheit**

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

- **Klassifikationsgesichtspunkte**

Bearbeitung in Ein- / Mehrprozessorsystemen

Bearbeitung ohne / mit Prozessorentzug

Deterministische / probabilistische Modelle

Echtzeitbedingungen

Scheduling – Deterministische Modelle

- **Modellannahmen**

Gegeben:

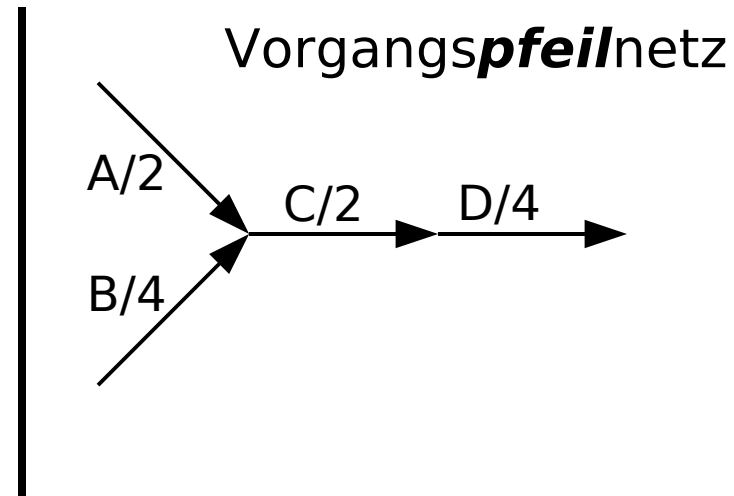
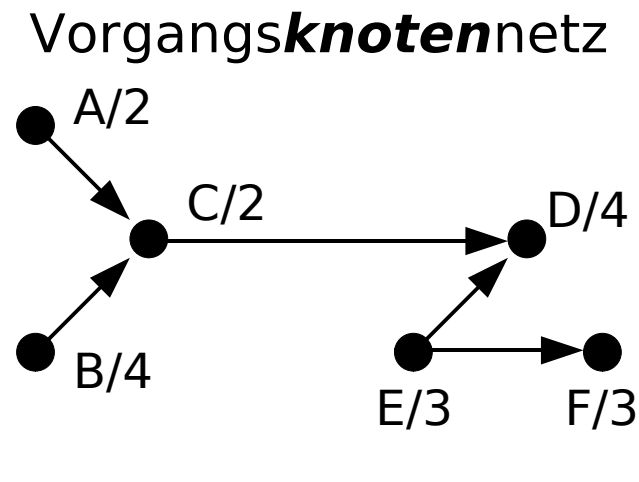
$J = \{J_1, \dots, J_n\}$ Menge von Jobs
 $R \subseteq J \times J$ Präzedenzrelation
 $t: J \rightarrow \mathbb{R}^+$ Abbildung, wobei $t(J_i) =: t_i$

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

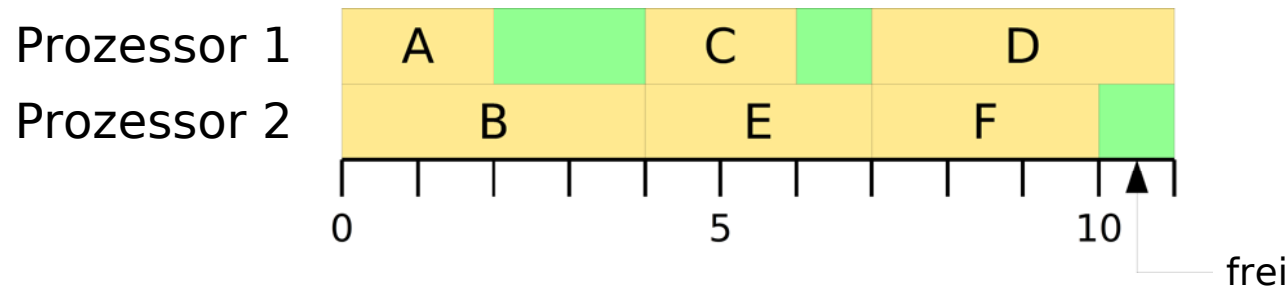
Anwendungsbereich: (annähernd) konstantes Aufgabenprofil

Scheduling – Deterministische Modelle

- Graphische Darstellung



Möglicher Ablaufplan bei zwei Prozessoren:



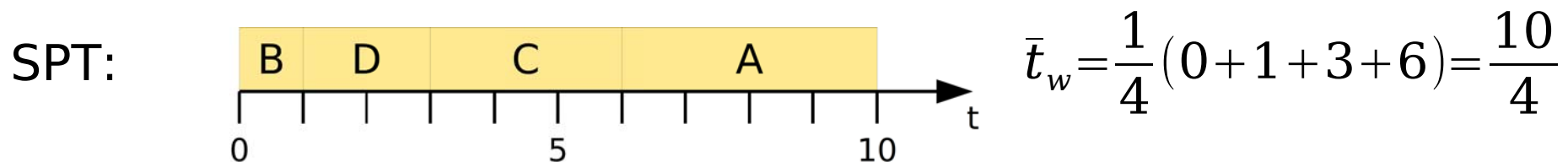
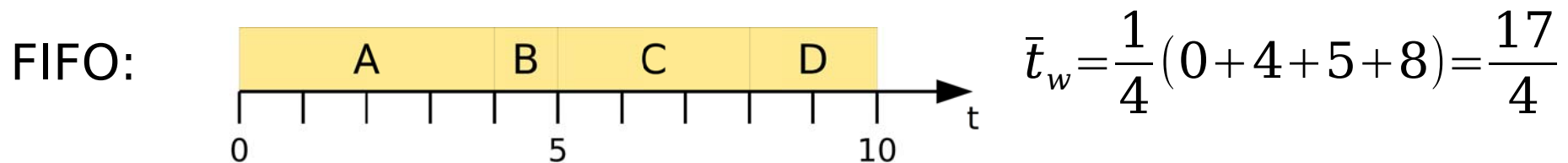
Scheduling in 1-Prozessor-Systemen ohne Entzug

- **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!

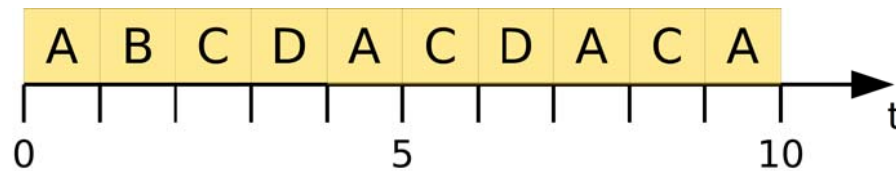
Scheduling in 1-Prozessor-Systemen mit Entzug

- **RR „Round Robin“**

Prozeßwechsel mit konstantem Zeitquant Q

Beispiel:

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



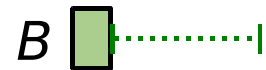
$$\bar{t}_w = \frac{1}{4} (6 + 1 + 6 + 5) = \frac{18}{4}$$

Problem: Größe von Q

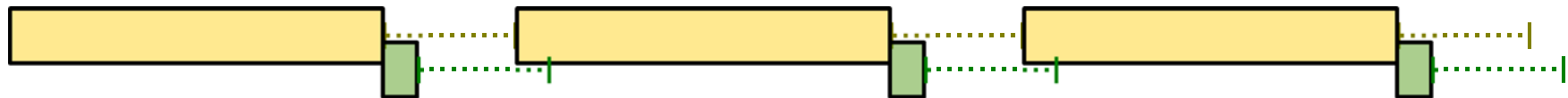
Scheduling in 1-Prozessor-Systemen mit Entzug

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

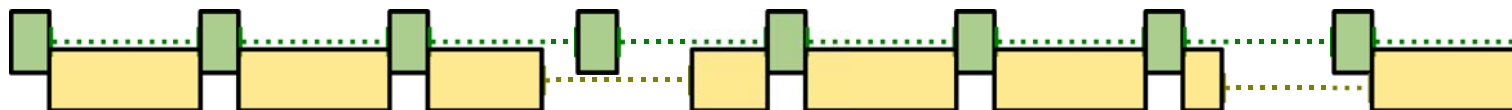
Antwortzeit und Durchsatz abhängig von Priorität



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 mit Prozessorentzug**

$R = \emptyset$:

$$t_{g,\min} = \max\left(\frac{1}{m} \cdot \sum_{i=1}^m t_i, \max_i(t_i)\right)$$

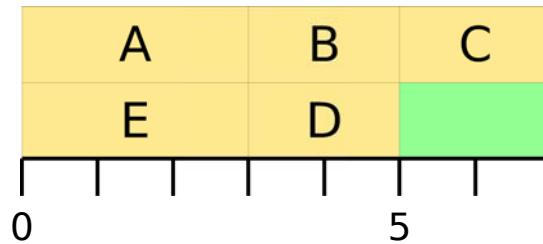
Scheduling in Mehrprozessor-Systemen

- Beispiele**

$m = 2$

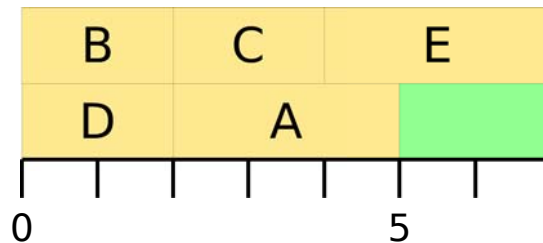
J_i	A	B	C	D	E
t_i	3	2	2	2	3

LPT:



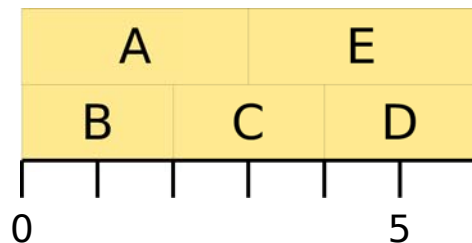
$$t_g = 7 \quad \bar{t}_w = \frac{11}{5}$$

SPT:



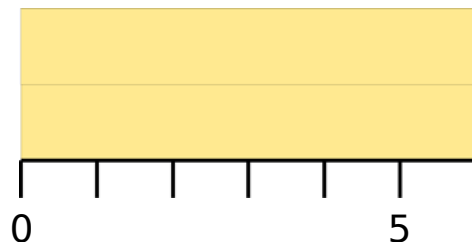
$$t_g = 7 \quad \bar{t}_w = \frac{8}{5}$$

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



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

Entzug:

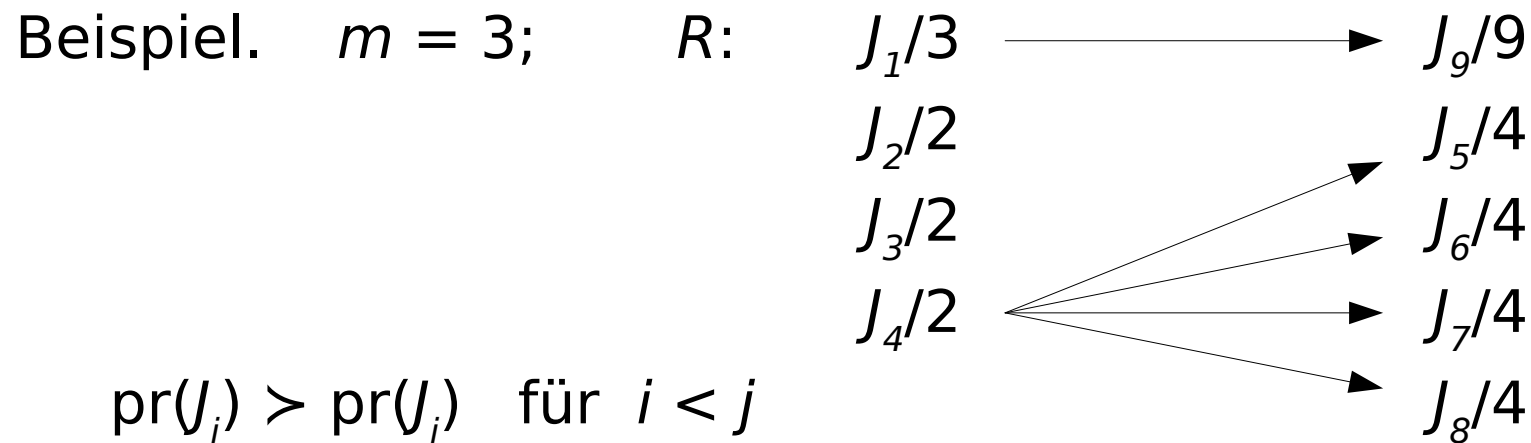


$$t_g = 6 \quad \bar{t}_w = \frac{8}{5}$$

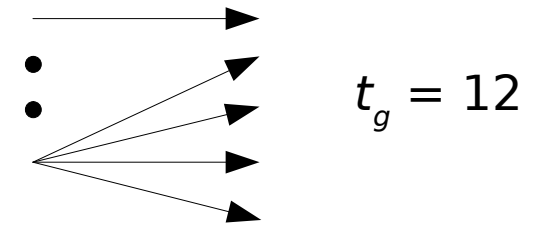
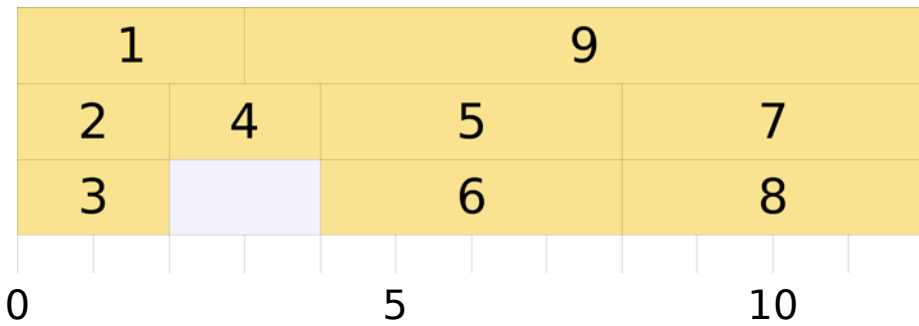
Mehrprozessor-Anomalien

Für eine Menge von Jobs sei bei gegebenen Bearbeitungszeiten, Prioritätszuordnungen, Prozessoranzahl und Präzedenzrelation ein Ablaufplan gefunden worden. Dann kann eine der folgenden Aktionen zu einer **Verlängerung** der Gesamtbearbeitungszeit t_g führen:

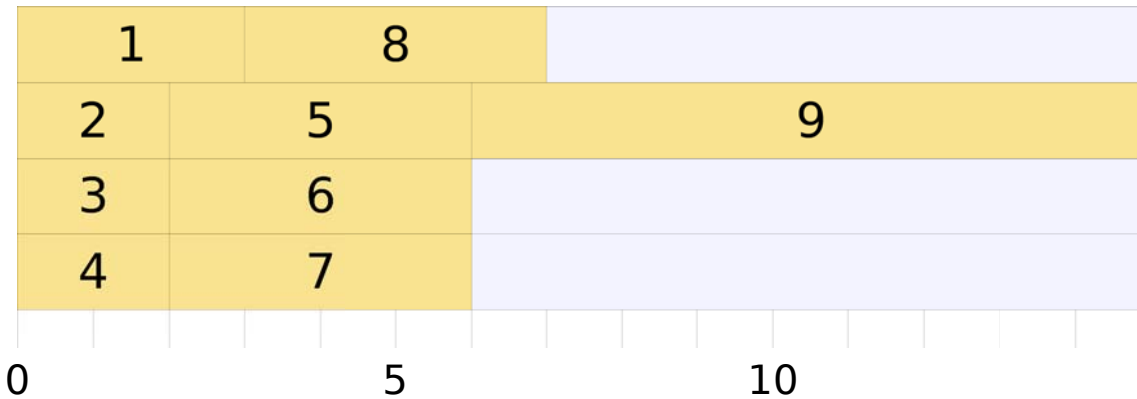
- Erhöhung der Prozessoranzahl
- Verringerung der Bearbeitungszeiten
- Abschwächung der Präzedenzrelation
- Veränderung der Prioritäten.



Mehrprozessor-Anomalien

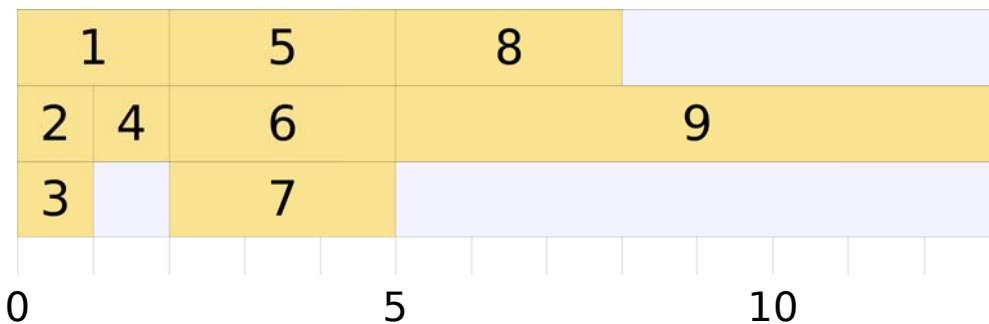


Erhöhung Anzahl Prozessoren:



$$t_g = 15$$

Verkürzen aller Bearbeitungszeiten um 1 Einheit:



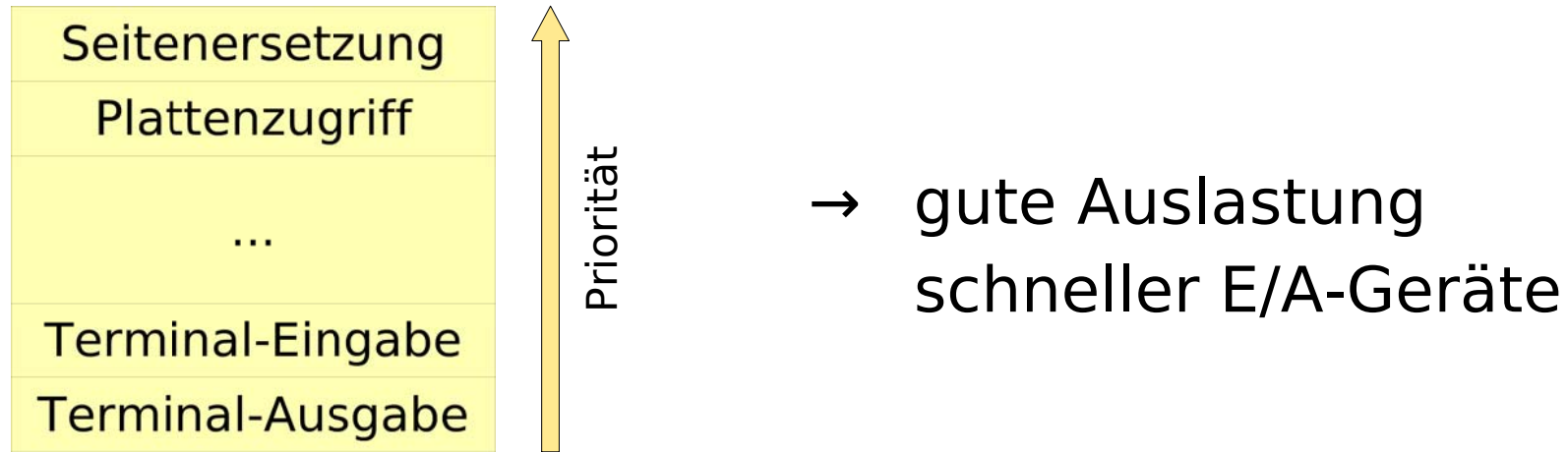
$$t_g = 13$$

Aufheben der Abhängigkeiten $J_4 \rightarrow J_5$, $J_4 \rightarrow J_6$: $t_g = 16$

Umordnen der Priorität: $J_1 J_2 J_4 J_5 J_6 J_3 J_9 J_7 J_8$: $t_g = 14$

Fallstudie: Unix (konzeptionell)

- **Statische Prioritäten im Kern-Modus**



- **Dynamische Prioritäten im Nutzer-Modus**

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

→ Bevorzugung interaktiver Anwendungen

Fallstudie: Linux

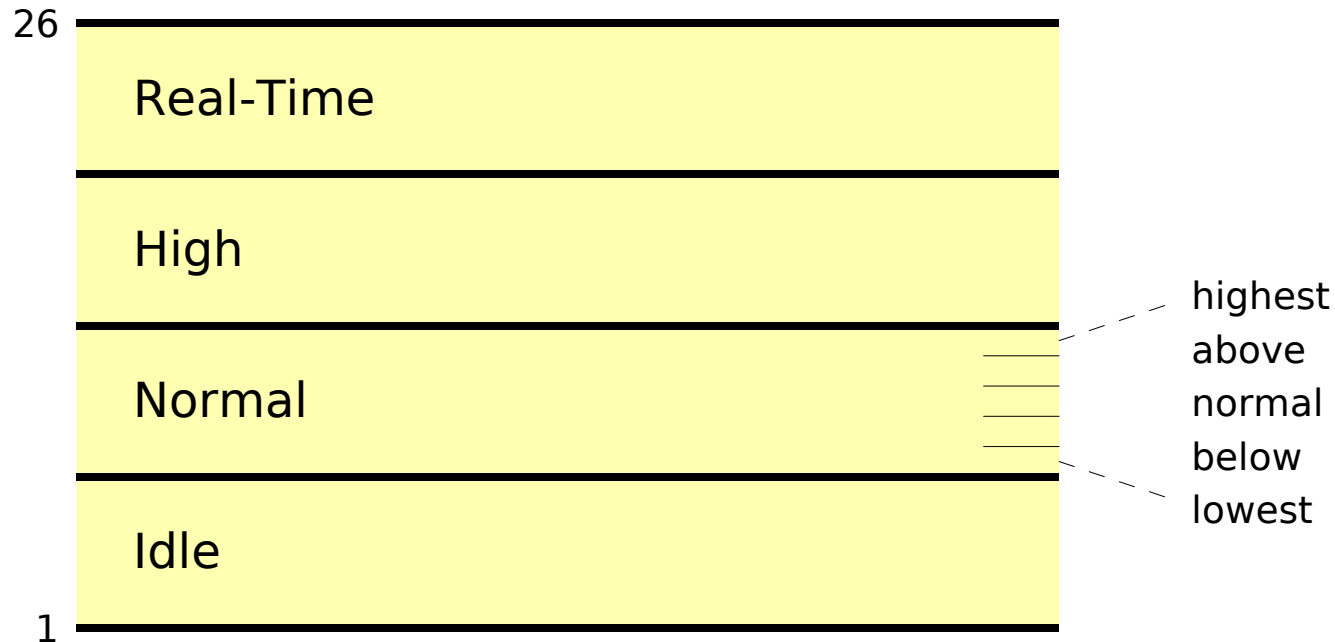
- **Prioritätsklassen / Scheduling-Strategien**

- SCHED_FIFO keine Verdrängung!
 - SCHED_RR zeitscheiben-basiert
 - SCHED_OTHER dialogorientiert
- } Echtzeit
- Time-sharing

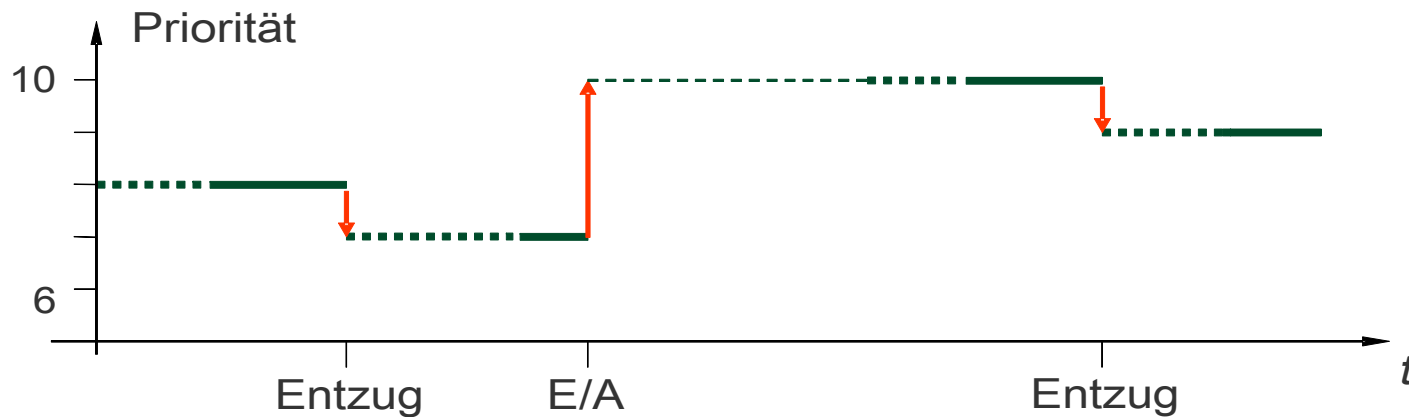
„Prioritäts-Bonus bzw. -Malus“
für E/A-intensive bzw. rechenintensive Jobs
Basis: Messung der Blockierzeiten

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)
- d Zeitschranke, Frist (deadline)
- r Freigabezeit, Bereitzeit (release time)

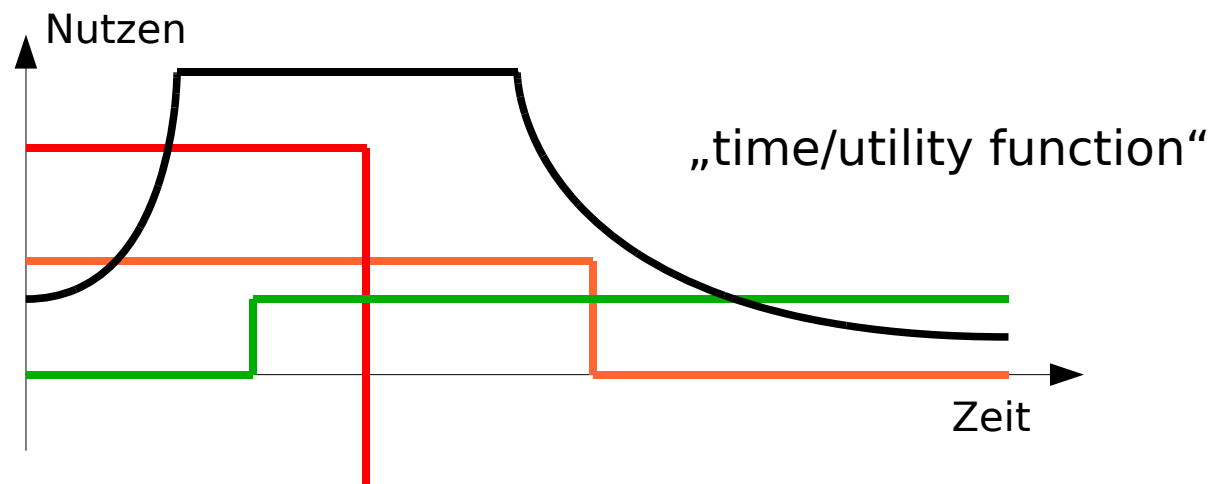
- **Task**

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): Einhaltung aller Zeitschranken

- **Scheduling**

- **Einplanung:** Vorgehen (Algorithmus), das bei gegebener Taskbeschreibung einen Ablaufplan bestimmt

- **Prozessor-Zuordnung:** Auswahl eines Jobs durch Scheduler des Systems

- **Ressource (Betriebsmittel)**

Prozessor, i.d.R. entziehbar (Kosten!)

weitere, nicht entziehbare Ressourcen

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 \mathcal{C} von Verfahren:
erzeugt für jede Taskmenge T einen ausführbaren Ablaufplan, sofern T überhaupt mit einem Verfahren aus \mathcal{C} eingeplant werden kann

Echtzeit-Scheduling – Grundbegriffe

- **Klassifikationsgesichtspunkte**
 - Scheduling für Jobnetze | periodische Tasks
 - Scheduling für periodische Tasks:
 - zeitgesteuert (time driven)
 - ereignisgesteuert (event driven)
 - statische | dynamische Prioritäten
 - nicht-periodische Tasks
 - Nutzung weiterer, nicht entziehbarer Betriebsmittel
 - Entziehbarkeit
 - Ein-/Mehrprozessor-Systeme

Echtzeit-Scheduling – Modellannahmen

- **Deterministisches Modell**

jede Task T_i ist periodische Folge von Jobs, Periode p_i

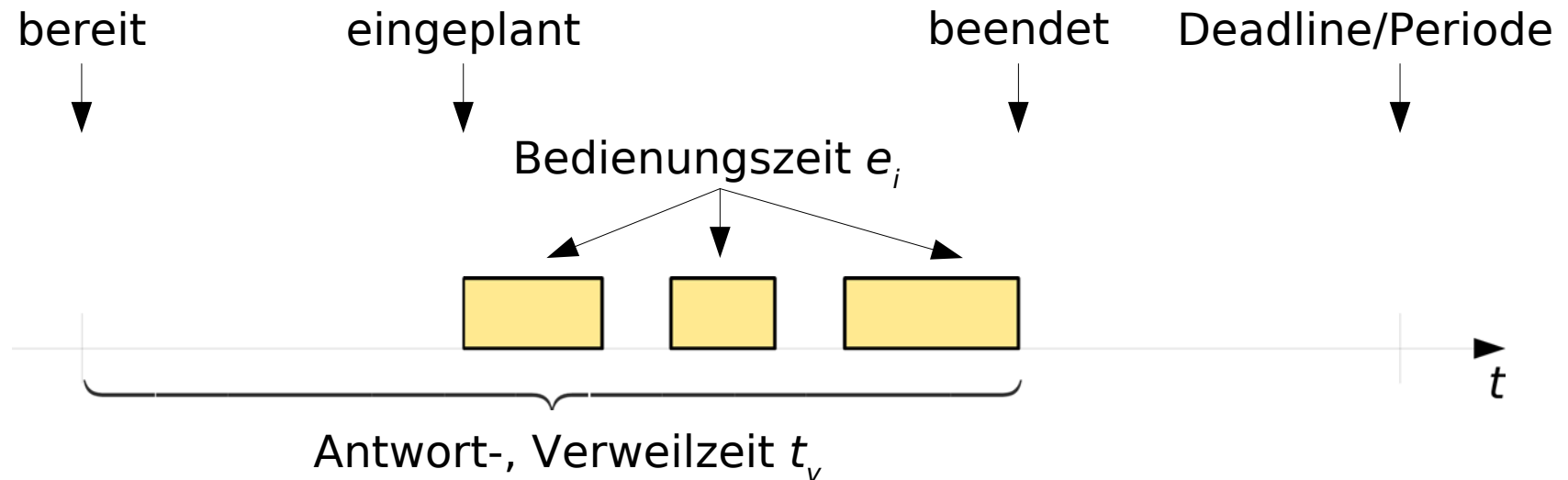
Periode ist zugleich Zeitschranke (relative Deadline d_i)

Bearbeitungszeit e_i ist konstant

Prozessor ist entziehbar

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

System-Scheduling prioritätsbasiert



Echtzeit-Scheduling – weitere Begriffe

- **Hyperperiode**

$$H = \text{kgV}(p_1, \dots, p_n)$$

- **Harmonische Perioden**

$$p_i \leq p_j \rightarrow p_i \mid p_j \quad \forall i, j = 1, \dots, n$$

- ***Nicht-periodische Tasks***

- *aperiodisch: Mindestabstand*
- *sporadisch: beliebiger Abstand*

Echtzeit-Scheduling – Verfahren

- **RMS** „Rate Monotonic Scheduling“

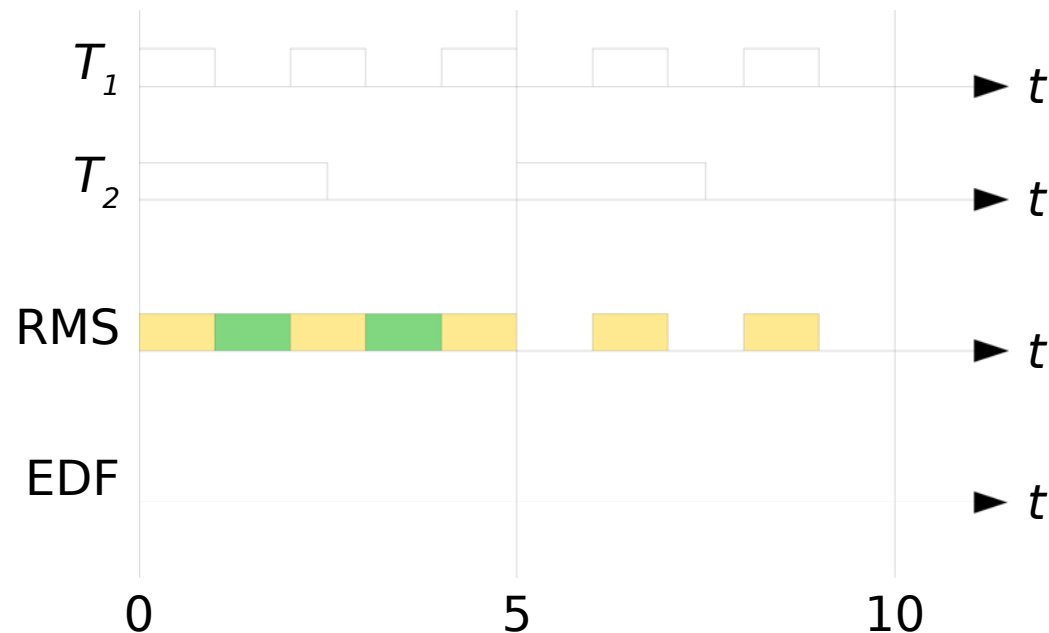
Statische Taskpriorität proportional zu Ankunftsrate

- **EDF** „Earliest Deadline First“

Task mit nächstgelegener Deadline hat aktuell höchste Priorität

- **Beispiel**

T_1 hat höhere
Priorität als T_2



Echtzeit-Scheduling – Eigenschaften

- **Optimalität bzgl. Einplanbarkeit**

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

Optimalität von RMS:

Taskmenge T „irgendwie“ einplanbar $\rightarrow T$ mit RMS einplanbar

- **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)

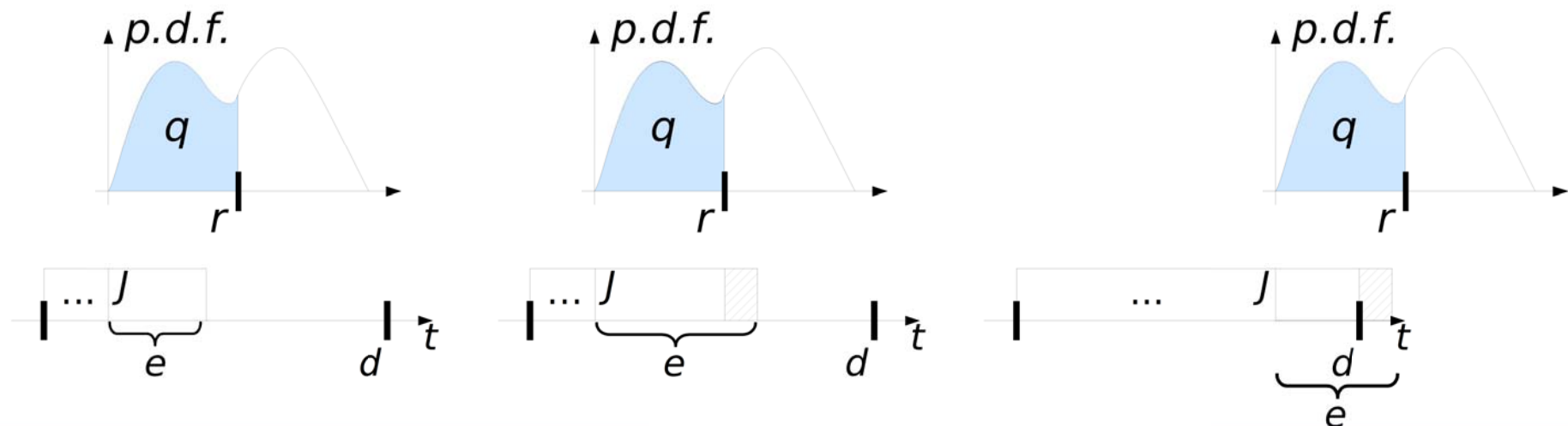
bei EDF genau dann, wenn $\eta = \sum_{i=1}^n \frac{e_i}{p_i} \leq 1$

Echtzeit-Scheduling – Ausblick: QAS

• Motivation

- Architekt von DROPS – Dresden Real-Time Operating System
Scheduling mit festen Prioritäten und Reservierungsprioritäten
- Arbeitslast
periodische, unabhängige Tasks mit „weichen“ Echtzeitbedingungen
stark schwankender Ressourcenbedarf
- Hintergrund
Imprecise Computations, Statistical Rate Monotonic Scheduling

• Idee



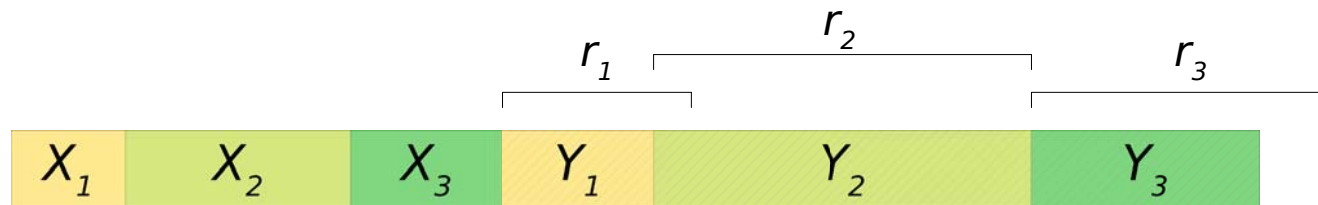
Echtzeit-Scheduling – Quality-Assuring Scheduling

- **Ausgangspunkt**

Aufteilen der Jobs einer periodischen $Task T_i$ in **Pflichtteil** X_i und **Wahlteil** Y_i (Qualität q_i : Anteil vollständig bearbeiteter Wahlteile)

- **Vorgehen**

Scheduling auf der Basis statischer Reservierungsprioritäten



$$r_i = \min \left(r \in \mathbb{R} \mid P \left(Y_i \leq r \wedge \sum_{j=1}^n X_j + \sum_{k=1}^{i-1} \min(Y_k, r_k) + Y_i \leq d \right) \right) \quad (*)$$

- **Admission**

(1) $\sum_{i=1}^n w_i \leq d$ d : gemeinsame Deadline
 w_j : WCET des Pflichtteils von T_i

(2) Das Gleichungssystem (*) ist lösbar mit $r_i \leq d \quad \forall i = 1, \dots, n$

Scheduling - Probabilistische Modelle

- **Modellannahmen**

Zu zufälligen Zeiten A_i (Ankunftszeit) treffen Anforderungen an den Scheduler in zufälligem Umfang B_i (geforderter Bedarf nach dem betrachteten Betriebsmittel, Bedienungszeit) ein.
Anwendungsbereich: Interaktiver Betrieb, verteilte Systeme

- **Mathematisches Instrumentarium**

Bedienungstheorie

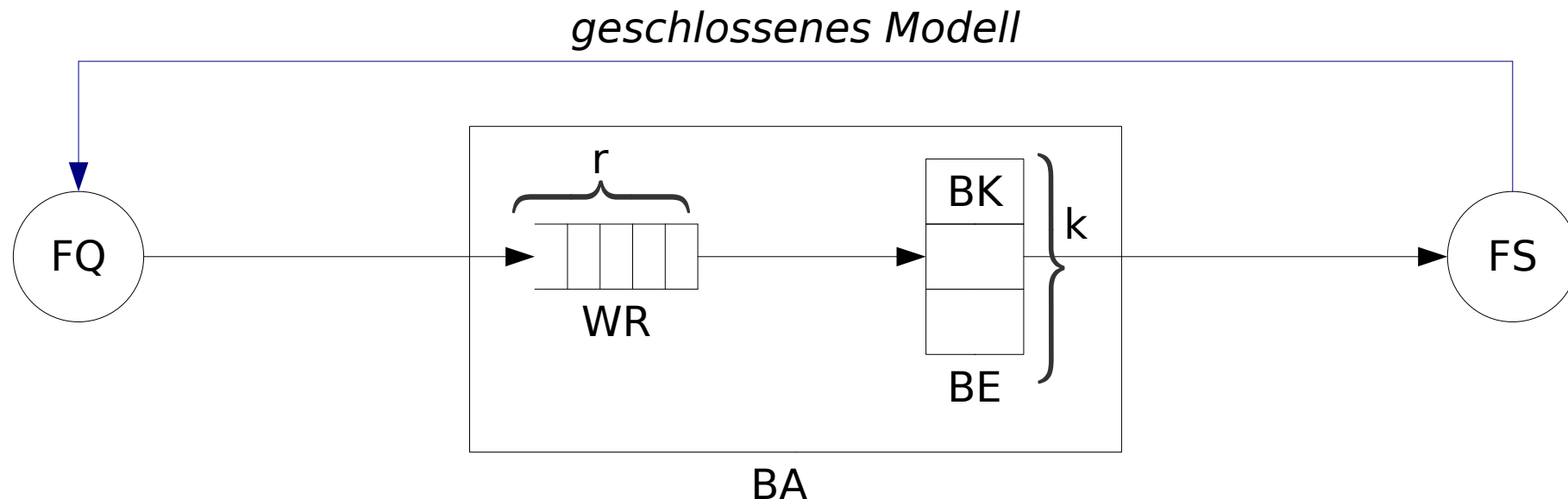
Warteschlangentheorie

Теория массого обслуживания

- **Historie**

1908	ERLANG	Kopenhagener Telegrafenamnt
1910	PALM	Mehrmaschinenbedienung
1940 ...1970	KANTOROWITSCH, GNEDENKO, KOWALENKO	
1965 ...1980	KLEINROCK, COFFMAN, DENNING	

Bedienungssysteme – Strukturbeschreibung



FQ: Forderungsquelle

FS: Forderungssenke

BA: Bedienungsanlage

WR: Warteraum

BE: Bedienungseinrichtung

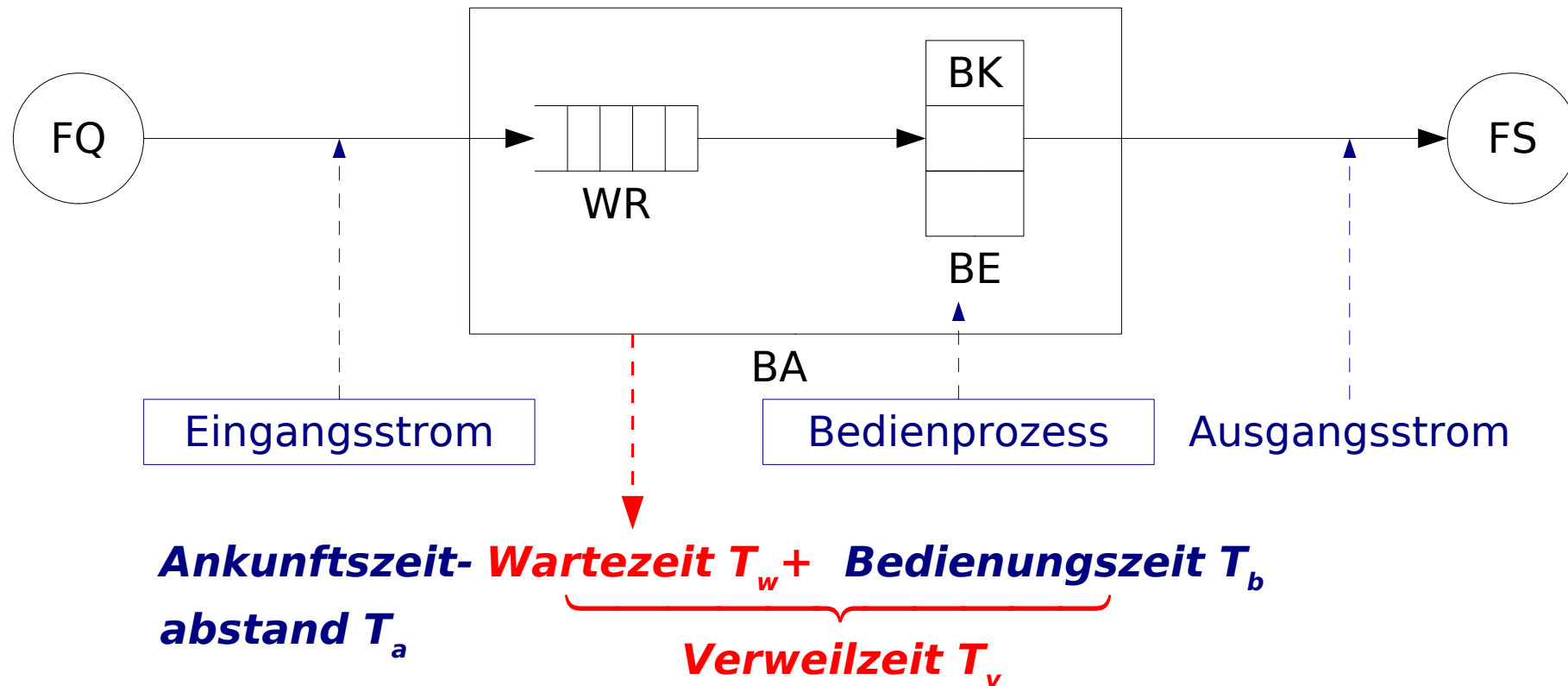
BK: Bedienungskanal

• **Strukturbeschreibung**

- r : Größe des Warteraums
- k : Anzahl Bedienungskanäle

- Bedienungsstrategie !
- offen/geschlossen

Bedienungssysteme – Lastbeschreibung



- **Lastbeschreibung**

- Ankunftsprozeß

- Bedienungsprozeß

Bedienungssysteme – Prozeßbeschreibung

- **Ankunftsprozeß**



N_a zufällige Anzahl der pro Zeiteinheit eintreffenden Ford.

T_a zufällige Zeitdauer zwischen dem Eintreffen zweier Ford.

oft: *Poissonscher Ankunftsprozeß mit der Rate $\lambda > 0$:*

$$P(N_a = i) = \frac{\lambda^i}{i!} \cdot e^{-\lambda}, i \in \mathbb{N}$$

POISSON-Verteilung

$$P(T_a \leq t) = \begin{cases} 1 - e^{-\lambda t} & t \geq 0 \\ 0 & t < 0 \end{cases}$$

Exponentialverteilung

$$\boxed{E(N_a) = \lambda = \frac{1}{E(T_a)}}$$

Eingangsstromintensität,
Ankunftsrate

Bedienungssysteme - Prozeßbeschreibung

- **Bedienprozeß**

T_b zufällige Dauer der Bedienung einer Forderung

oft: T_b exponentiell verteilt, Parameter μ

- **Weitere Verteilungen**

ERLANG-Verteilung k -ter Ordnung, $k \in \mathbb{N}^+$

$$f(t) = \frac{\alpha^k t^{k-1}}{(k-1)!} \cdot e^{-\alpha t} \quad (t \geq 0)$$

Hypo-Exponentialverteilung

$$f(t) = \frac{\alpha_1 \alpha_2}{\alpha_1 - \alpha_2} (e^{-\alpha_1 t} - e^{-\alpha_2 t}) \quad (t \geq 0)$$

Bedienungssysteme – Kurzbeschreibung

- **KENDALLSche Notation**

A/B/k/r



Warteraumkapazität, $r \geq 0 / \infty$

Anzahl der Bedienungskanäle, $k > 0$

Bedienprozess

Ankunftsprozess

Häufige Werte für A,B:

- M POISSON-Prozess / exponentialvert. Zeit
- E_k ERLANG-Verteilung k -ter Ordnung
- D deterministische Verteilung
- G allgemeiner Fall

- **Bedienungsstrategie**

FIFO, LIFO, Random, Prioritäten, ...

- **Geschlossenes Modell**

n : Anzahl der Forderungen (statt r)

Bedienungssysteme – Bewertungsgrößen

- „Elementare“ Bewertungsgrößen

$E(T_a)$ mittlerer Ankunftszeitabstand

$E(T_b)$ mittlere Bedienzeit

$\lambda = \frac{1}{E(T_a)}$ Ankunftsrate

$\mu = \frac{1}{E(T_b)}$ Bedienrate

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

Verkehrswert
(**Be**-lastung,
Angebot)

Forderung: $\rho < k$ für nicht-kritischen Zustand

Bedienungssysteme – Bewertungsgrößen

- **Bewertungsgrößen i. e. S.**

N_w Warteschlangenlänge

N_b Anzahl der aktuell bedienten Forderungen

$N_v = N_w + N_b$ Anzahl der im System befindlichen Forderungen

T_w Wartezeit

$T_v = T_w + T_b$ Verweilzeit

Für GI/G/k/∞-Systeme gilt:

$$E(T_v) = E(N_v) \cdot E(T_a) \quad (\text{LITTLE})$$

Bedienungssysteme – Bewertungsgrößen

- **Grundlage**

Zustandswahrscheinlichkeiten $p_i = P(N_v = i) \quad i \in \mathbb{N}$

- **Allgemeines Vorgehen**

Ermitteln des Zustandsgraphen

Aufstellen der Zustandsgleichungen

Lösen des Gleichungssystems

Berechnen der Zustandswahrscheinlichkeiten

Berechnen der Bewertungsgrößen

M/M/1/∞-System

- **Struktur**

Eingangstrom: POISSON-Prozess, Intensität λ

Bedienungszeit: exponentiell verteilt, Parameter μ

Bedienungsstrategie: FIFO, LIFO, gleichverteilt; ohne Entzug

Verkehrswert $\rho = \frac{\lambda}{\mu} < 1$!

- **Zustandswahrscheinlichkeiten**

$$p_i = P(N_v = i) = \rho^i (1 - \rho) \quad i \in \mathbb{N}$$

M/M/1/∞-System – Bewertungsgrößen

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

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

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

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

$$E(T_w^2) = \begin{cases} \frac{2\rho}{\mu^2(1-\rho)^2} & \text{FIFO} \\ \frac{2\rho}{\mu^2(1-\rho)^3} & \text{LIFO} \\ \frac{2\rho}{\mu^2(1-\rho)^2(1-\frac{\rho}{2})} & \text{RANDOM} \end{cases}$$

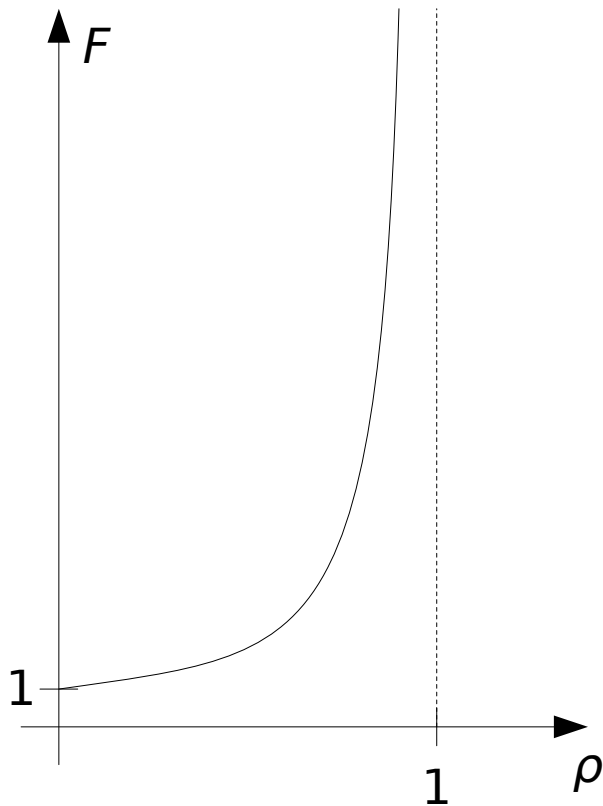
Wartezeit-Verteilung für M/M/1/∞-FIFO:

$$P(T_w \leq t) = 1 - \rho e^{-(\mu-\lambda)t} \quad (t \geq 0)$$

M/M/1/∞-System – Systemverhalten

- **Verweilzeitfaktor**

$$F = \frac{E(T_v)}{E(T_b)} = \frac{1}{1-\rho}$$



ρ	0,4	→	0,44
F	1,67	→	1,79

ρ	0,9	→	0,94
F	10	→	16,7

ρ	0,9	→	0,99
F		→	

Weitere Standardmodelle

- **M/G/1/∞-System**

$$E(T_w) = \frac{\lambda \cdot E(T_b^2)}{2(1-\rho)}$$

- **M/M/k/∞-System**

$$E(N_w) = \frac{\rho^{k+1}}{(k-1)!(k-\rho)^2} \cdot p_0 \quad \eta = \frac{\rho}{k}$$

$$p_0 = \left[\sum_{i=0}^{k-1} \frac{\rho^i}{i!} + \frac{\rho^k}{(k-1)!(k-\rho)} \right]^{-1}$$

- **M/G/k/0-System**

$$p_V = p_k = \frac{\rho^k}{k!} \cdot p_0, \quad p_0 = \left[\sum_{i=0}^k \frac{\rho^i}{i!} \right]$$

Bewertung von Scheduling-Verfahren

- **Bewertungsmaß und Strategien**

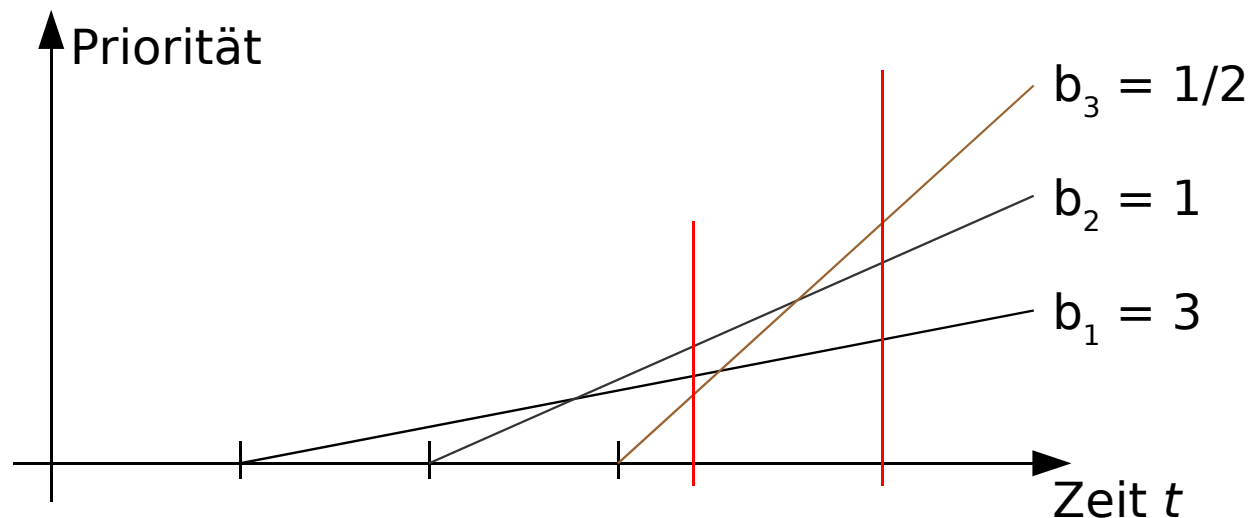
Abhängigkeit der mittleren Wartezeit $E(T_w)$ vom Bedienzeitwunsch b der Forderungen für die Strategien (ohne Entzug)

FIFO - LIFO - Random

SJN (SPT)

HRN „Highest Response Ratio Next“

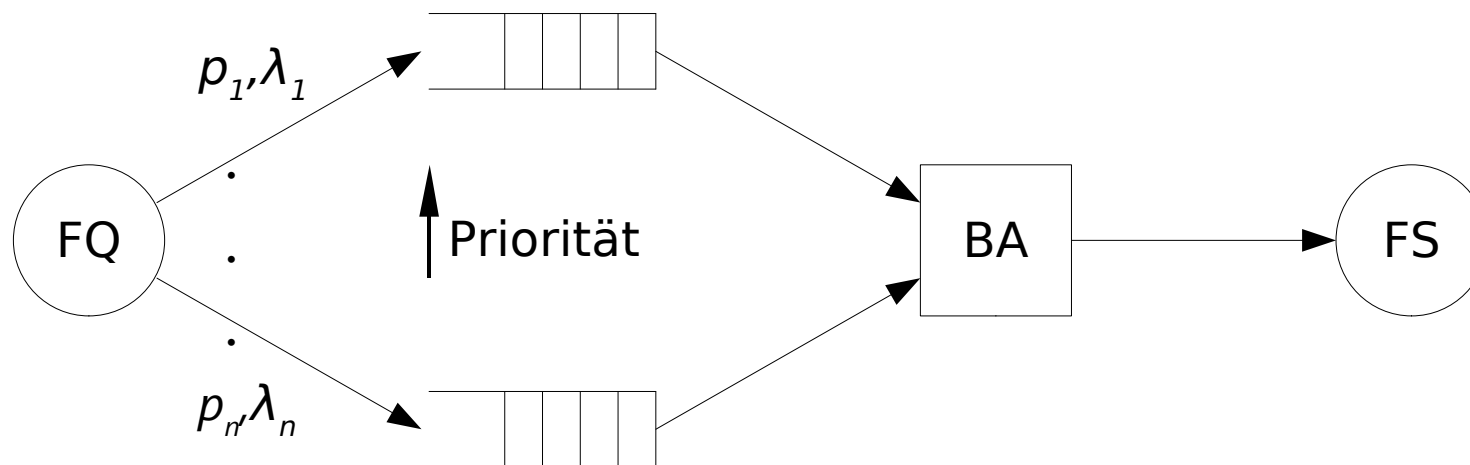
$$\text{Priorität} = \frac{\text{aktuelle Verweilzeit}}{\text{Bedienzeitwunsch}} = \frac{t - t_i}{b_i}$$



Bewertung von Scheduling-Verfahren

FEP *Fixed External Priorities*

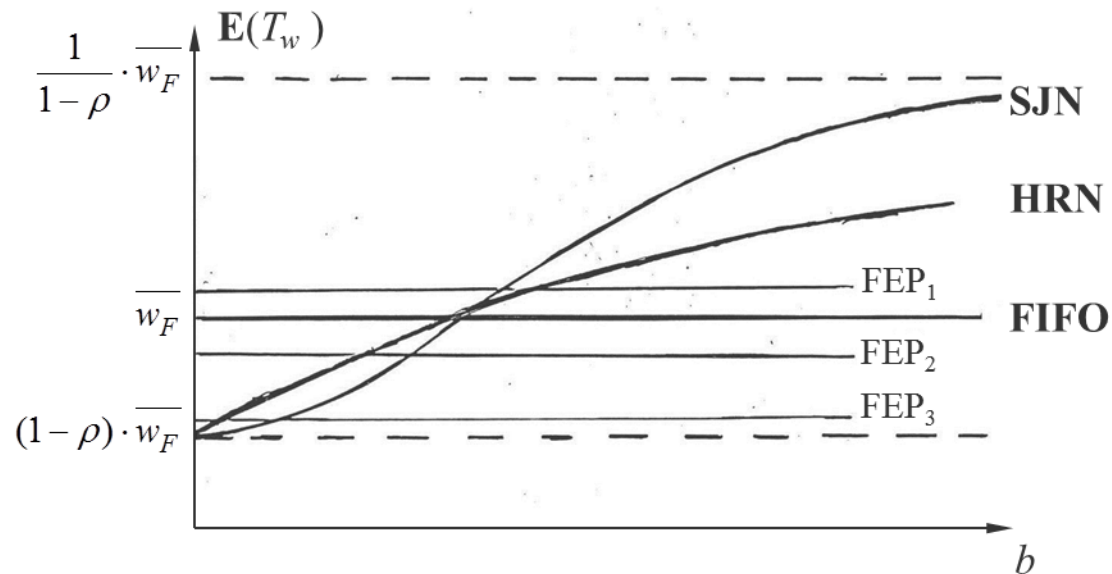
Aufträge werden in Prioritätsklassen eingeordnet
innerhalb einer Klasse FIFO
Bedienrate klassenunabhängig



- **Grundlage:** M/G/1/ ∞ -System mit dynamischen Prioritäten

Bewertung von Scheduling-Verfahren

- **Ergebnisse**



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: Speicherverwaltung

- **Bewertung von Speicher-Zuteilungsstrategien**

- ***Segmentierter Speicher***

Freispeicherlisten → externe Fragmentierung

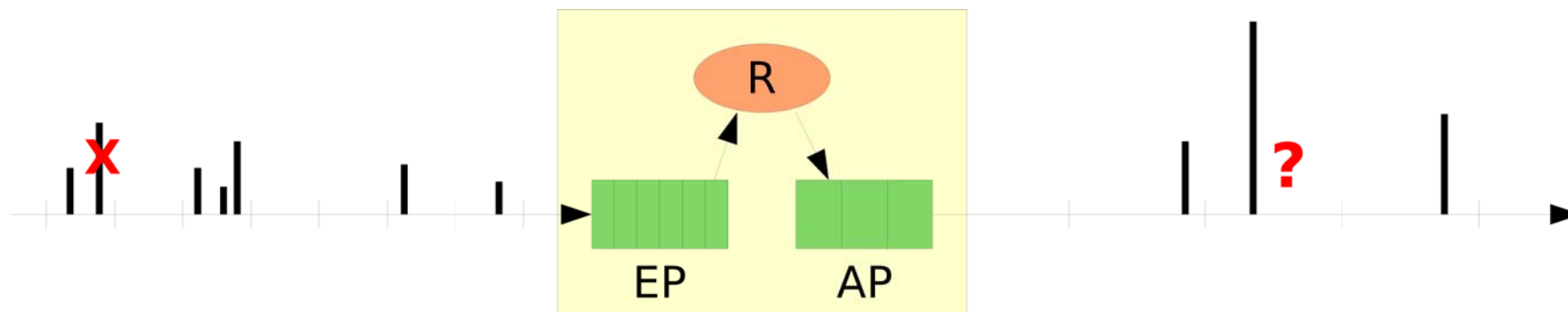
Auffüllen und Kompaktifizieren → Aufwand

- ***Seitenorientierte Systeme***

Arbeitsmengen → Fenstergröße

statische Ersetzungsstrategien → Seitenfehlerrate

- **Pufferdimensionierung !**



Ausblick: Magnetplatte – Positionierabstand

k : Spuranzahl

D : Positionierabstand

Forderungsstrom POISSONSCH, Intensität λ

- **Bedienungsstrategie FIFO**

$$E D_{FIFO} = \frac{k^2 - 1}{3k} \sim \frac{k}{3}$$

- **Bedienungsstrategie SSTF (Shortest Seek Time First)**

$$E D_{SSTF} \sim \frac{k}{n+1}$$

n : Anzahl der wartenden Aufträge

- **Bedienungsstrategie SCAN (*Fahrstuhl-Algorithmus*)**

$$E D_{SCAN} = \frac{k}{\lambda \tau} (1 - e^{-\lambda \tau})$$

τ : Zeit zum Überfahren der Platte

Ausblick: Scheduling

- **Scheduling**

Leistungsanalyse mittels Bedienungsnetzen

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**



***„Quantitative Methoden
der Betriebssysteme-Konstruktion“***

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	17
3. Scheduling in Echtzeitsystemen	20
3.1. Grundlagen	20
3.2. Verfahren	26
4. Scheduling - Probabilistische Modelle	30
4.1. Grundbegriffe der Bedienungstheorie	31
4.2. Das M/M/1/∞-System	39
4.3. Anwendung: Bewertung von Scheduling-Strategien	43
5. Ausblick	46