

Aufgabe 2.1: Scheduling-Handsimulation (Tafelübung)

Es werden in einem Ein-Prozessor-System Prozesse wie in Tabelle 1 beschrieben gestartet:

Prozess	A	B	C	D
Ankunftszeitpunkt	0	3	4	8
Dauer	5	3	8	4
Priorität	4	1	2	3

Tabelle 1: Prozesse eines Systems mit einer CPU und einem Thread

a) Simulieren¹ Sie folgende Scheduling-Verfahren für die Prozesse aus Abbildung 1:

- FCFS,
- PRIO-NP,
- SRTN,
- RR mit $\tau = 2$.

Bei gleichen Prioritäten verwenden Sie das FCFS-Verfahren.

b) Simulieren Sie das MLF Scheduling-Verfahren für die Prozesse aus Abbildung 1 an der Tafel mit $\tau_i = 2^i$ ($i = 0, 1, \dots$).

c) Berechnen Sie für jedes der verwendeten Verfahren

- die Warte- und Antwortzeit *jedes* Prozesses sowie
- die mittlere Warte- und Antwortzeit des gesamten Systems.

Lösung:

a) **Anmerkung:** Die tiefgestellte Zahl in den Tabellen der Lösung stellt immer die Restlaufzeit des Prozesses vor dem aktuellen Zeitschritt dar.

FCFS:

Zeit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
CPU	A_5^1	A_4^1	A_3^1	A_2^1	A_1^1	B_3^2	B_2^2	B_1^2	C_8^3	C_7^3	C_6^3	C_5^3	C_4^3	C_3^3	C_2^3	C_1^3	D_4^4	D_3^4	D_2^4	D_1^4
Queue				B_3^2	B_2^2	C_8^3	C_8^3	C_8^3	D_4^4	D_4^4	D_4^4	D_4^4	D_4^4	D_4^4	D_4^4	D_4^4				

Anmerkung: Die hochgestellte Zahl gibt die Startreihenfolge der Prozesse an.

¹<https://dos-group.github.io/SysprogInteract/>

Prio-NP:

Zeit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
CPU	A_5^4	A_4^4	A_3^4	A_2^4	A_1^4	C_8^2	C_7^2	C_6^2	C_5^2	C_4^2	C_3^2	C_2^2	C_1^2	D_4^3	D_3^3	D_2^3	D_1^3	B_3^1	B_2^1	B_1^1
Queue				B_3^1	C_8^1	B_3^1	B_3^1	B_3^1	D_4^3	D_4^3	D_4^3	D_4^3	D_4^3	B_3^1	B_3^1	B_3^1	B_3^1			

Anmerkung: Die hochgestellte Zahl gibt den Prioritätswert des Prozesses an. Je größer dieser Wert, desto „wichtiger“ ist der Prozess.

SRTN:

Zeit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
CPU	A_5	A_4	A_3	A_2	A_1	B_3	B_2	B_1	D_4	D_3	D_2	D_1	C_8	C_7	C_6	C_5	C_4	C_3	C_2	C_1
Queue				B_3	B_3	C_8	C_8	C_8	C_8	C_8	C_8	C_8								

RR

Zeit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
CPU	A_5	A_4	A_3	A_2	B_3	B_2	C_8	C_7	A_1	B_1	D_4	D_3	C_6	C_5	D_2	D_1	C_4	C_3	C_2	C_1
Queue				B_3	C_8	C_8	A_1	A_1	B_1	D_4	C_6	C_6	D_2	D_2	C_4	C_4				

Checkliste - Wie werden Prozesse gescheduled:**1. Kommt ein neuer Prozess an?**

Wenn ja, so wird er am Ende der Queue eingefügt.

2. Ist das Quantum abgelaufen?

Wenn ja, so wird der Prozess der die CPU belegt am Ende der Queue eingefügt.

3. Ist die CPU ungenutzt?

Wenn ja, so wird in der Queue nachgeschaut, ob Prozesse warten. Dem Obersten (ergo dem am längsten Wartenden) wird die CPU für die Länge eines Quantums zur Verfügung gestellt.

b) MLF:

Zeit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
CPU	A_5^0	A_4^1	A_3^1	B_3^0	C_8^0	B_2^1	B_1^1	C_7^1	C_6^1	D_4^0	D_3^1	D_2^1	A_2^2	A_1^2	C_5^2	C_4^2	C_3^2	C_2^2	D_1^2	C_1^3
Level 0									D_4^0											
Level 1					B_2^1	C_7^1	C_7^1													
Level 2			A_2^2	A_2^2	A_2^2	A_2^2	A_2^2	A_2^2	A_2^2	A_2^2	C_5^2	C_5^2	C_5^2	D_1^2	D_1^2	D_1^2	D_1^2			
Level 3																				C_1^3

Anmerkung: Die hochgestellte Zahl gibt das aktuelle Level des Prozesses an.

Hier gilt im Prinzip die gleiche Checkliste, wie beim RR-Verfahren. Der Unterschied ist lediglich, dass neue Prozesse in die Queue in Level 0 gepusht werden und die Prozesse nach Ablauf der Zeitscheibe in die Queue des nächsten Levels gepusht werden. Der komplette Ablauf kann wie folgt beschrieben werden:

1. Kommt ein neuer Prozess an?

Wenn ja: Füge den Prozess ans Ende der Queue in Level 0 ein.

2. Ist das Quantum des laufenden Prozesses abgelaufen?

Wenn ja: Sei dieser Prozess derzeit in Level n; füge den Prozess ans Ende der Queue in Level n+1 ein.

3. Ist die CPU ungenutzt?

Wenn ja: Beginnend in Level 0 wird in den Queues aller Level nacheinander nachgeschaut,

ob Prozesse warten. Dem obersten Prozess (ergo dem am längsten wartenden Prozess) des kleinsten Levels wird die CPU für die Länge eines Quantums gemäß des Levels zur Verfügung gestellt.

c) **Anmerkungen:**

- Es gilt das gleiche wie in a)

	FCFS		PRIO-NP		SRTN		RR		MLF	
	WZ	AZ	WZ	AZ	WZ	AZ	WZ	AZ	WZ	AZ
System	14/4	34/4	5	10	10/4	30/4	5	10	25/4	45/4
A	0	5	0	5	0	5	4	9	9	14
B	2	5	14	17	2	5	4	7	1	4
C	4	12	1	9	8	16	8	16	8	16
D	8	12	5	9	0	4	4	8	7	11

Aufgabe 2.2: Scheduling-Verfahren

(Tafelübung)

a) Benennen Sie die aus der Vorlesung bekannten Scheduling-Verfahren und ordnen Sie diese nach

(a) Strategiealternativen:

- ohne/mit Verdrängung,
- ohne/mit Prioritäten und
- unabhängig/abhängig von der Bedienzeit (BZ).

(b) Betriebszielen:

- Effizienz/Durchsatz,
- Antwortzeit und
- Fairness.

Begründen Sie Ihre Entscheidungen für die Betriebsziele!

b) Zwischen welchen zwei Schedulingzielen bildet das HRRN-Verfahren einen Kompromiss?

Lösung:

a) Einordnen

(a) Strategiealternativen:

	ohne Verdrängung		mit Verdrängung	
	ohne Priorität	mit Priorität	ohne Priorität	mit Priorität
BZ-unabhängig	FCFS, LCFS	PRIO-NP	LCFS-PR, RR, MLF	PRIO-P
BZ-abhängig	SJN, HRRN		SRTN	

(b) Betriebsziele:

Effizienz/Durchsatz: (reine CPU-Last/Batchverarbeitung) alle nicht Verdrängenden (wegen geringerer CPU-Zeit-Verschwendung), Zeitscheibenverfahren für gleichmäßige Auslastung (Peripherie)

Antwortzeit: SJN, SRTN

Fairness: RR, HRRN

- b) Das HRRN-Verfahren bildet einen Kompromiss zwischen dem Ziel kürzere Prozesse zu bevorzugen (Proportionalität) und dem Ziel längere Prozesse nicht zu lange warten zu lassen (maximale Verspätung).

Aufgabe 2.3: Periodische Prozesse

(Tafelübung)

Die Firma „Pen&Pencil“ möchte einen neuartigen Stift auf den Markt bringen. Dieser soll speziell in Meetings eingesetzt werden können und folgende Funktionen bieten: A) Die Beschleunigung aufzeichnen, so dass Geschriebenes einfach digitalisiert werden kann, B) Diese Daten (aus einem Puffer) auf die enthaltene MicroSD-Karte zu schreiben und C) Geschriebenes sofort ohne Verzögerung auf entsprechenden Boards über eine drahtlose Verbindung übertragen. Hierbei ist es wichtig, dass diese Aufgaben ohne Verzögerung möglichst schnell (ohne Verletzung der Deadline) und zuverlässig ausgeführt werden. In der nachfolgenden Tabelle sind die beispielhaften Eckdaten einer solchen Benutzung dargestellt: Dauer der Aufgabe und Periode, die zeitgleich auch die Frist (Deadline) ist. Alle Prozesse starten zeitgleich bei $t = 0$.

Prozesse	Dauer (D)	Periode (P)
A	1	3
B	1	4
C	1	6

Tabelle 2: Prozesse

- a) Existiert für diese Prozesse ein zulässiger Schedule? Wird das notwendige Kriterium erfüllt?
- b) Wie könnte dieser aussehen? Geben Sie etwaige Leerzeiten an und markieren Sie die Hyperperiode.
- c) Ist die Prozessmenge mit Rate-Monotonic-Scheduling (RMS) einplanbar? Begründen Sie Ihre Antwort.
- d) Was passiert, wenn Prozess C eine Dauer von $D = 2$ hätte? Könnten die Prozesse trotzdem gemäß RMS eingeplant werden?
- e) Was passiert, wenn Prozess C eine Dauer von $D = 2,5$ hätte?

Lösung:

- a) Ja. Die notwendige Bedingung ist erfüllt: $\sum_{i=1}^n \frac{D_i}{P_i} = \left(\frac{1}{3} + \frac{1}{4} + \frac{1}{6}\right) = 0,75 \leq 1$.
- b) Die Hyperperiode beträgt $t_{HP} = \text{LCM}(P_i) = \text{LCM}(3, 4, 6) = 12$, also reicht es aus, nur diese darzustellen, da sich danach alles wiederholt.

	1	2	3	4	5	6	7	8	9	10	11	12	13
A	X			X			X			X			
B		X			X				X				
C			X					X					

- c) Ja, da die hinreichende Bedingung erfüllt wird: $\sum_{i=1}^n \frac{D_i}{P_i} = \left(\frac{1}{3} + \frac{1}{4} + \frac{1}{6}\right) = 0,75 \leq n \cdot (\sqrt[n]{2} - 1) \approx 0,7798$
- d) Die notwendige Bedingung ist weiterhin erfüllt: $\sum_{i=1}^n \frac{D_i}{P_i} = \left(\frac{1}{3} + \frac{1}{4} + \frac{2}{6}\right) \approx 0,92 \leq 1$
Die hinreichende Bedingung ist allerdings nicht mehr erfüllt. Dennoch lassen sich diese Prozesse mit RMS einplanen (siehe Abbildung).

	1	2	3	4	5	6	7	8	9	10	11	12	13
A	X			X			X			X			
B		X			X				X				
C			X			X		X			X		

- e) Die notwendige Bedingung ist weiterhin erfüllt: $\sum_{i=1}^n \frac{D_i}{P_i} = \left(\frac{1}{3} + \frac{1}{4} + \frac{2,5}{6}\right) = 1 \leq 1$ Die Prozesse lassen sich aber in diesem Falle nicht mehr mit RMS einplanen. Prozess C würde schon die erste Deadline verpassen. Dies ist gut an der Abbildung aus vorheriger Teilaufgabe abzulesen. Der Prozess C wird exakt zum Zeitpunkt seiner ersten Deadline beendet. Wenn der Prozess also noch länger läuft, kann diese Deadline nicht eingehalten werden.

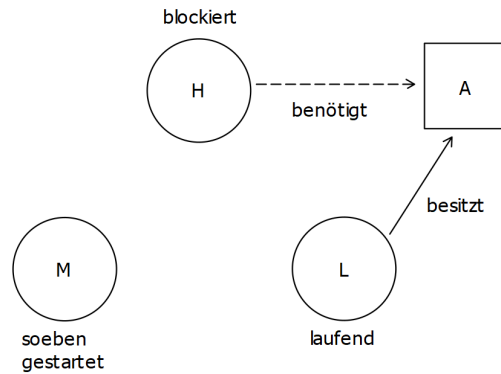


Abbildung 1: Prozesse eines Systems unter Verwendung eines verdrängenden Scheduling-Verfahrens mit Prioritäten

- Erklären Sie das Phänomen der *Prioritätsinversion*. Gehen Sie dabei davon aus, dass drei Prozesse *H*, *M* und *L* mit respektive hoher, mittlerer und geringer Priorität scheduled werden sollen. *H* und *L* benötigen das exklusive Betriebsmittel *A*, welches derzeit von *L* besetzt ist. Nun wird der Prozess *M* gestartet (momentane Situation wie in Abbildung 1) Was ist ein möglicher Lösungsansatz für dieses Problem? Erläutern Sie diesen kurz.
- Was sind die Unterschiede zwischen *Online*- und *Offline-Scheduling*? Gehen Sie dabei auch auf die benötigten Voraussetzung für beide ein.
- Was sind die Unterschiede und Gemeinsamkeiten von *Hard*- und *Soft-real-time-Systems*. Nennen Sie jeweils ein Beispiel.

Lösung:

- Der Prozess *H* ist der Prozess mit der höchsten Priorität, ist aber blockiert, da das benötigte Betriebsmittel *A* nicht vorhanden ist. Um *A* nutzen zu können muss *L* also das Betriebsmittel erst freigeben. Wenn nun ein Prozess *M* mit höherer Priorität als *L* eintrifft, wird *L* auf unbestimmte Zeit verdrängt. Somit kann Prozess *H* indirekt durch Prozess *M* (mit einer niedrigeren Priorität) nicht ausgeführt werden.

Lösung: Prioritätsvererbung. Solange ein Prozess *L* mit einer niedrigeren Priorität Betriebsmittel besetzt, die von Prozessen mit einer höheren Priorität benötigt werden, erbt Prozess *L* die höhere Priorität.

- Offline-Schedules benötigen alle Information über die Prozesse (Startzeit, Dauer, Priorität, etc.) und können vor Ablauf des eigentlichen Schedules schon konzipiert werden. Online-Schedules werden zur Laufzeit erstellt und arbeiten mit unvollständigen Information, haben dadurch aber auch keine Voraussetzungen.
- Beide Systeme sind Echtzeitsysteme in denen Sollzeitpunkte (Deadlines) im Schedule erfüllt werden müssen. In *Soft real-time systems* sind Verletzungen dieser allerdings im Gegensatz zu

Hard real-time systems tolerierbar und führen nur zu einem Qualitätsverlust und nicht zu einem Systemausfall.

Soft real-time systems: Internet-Telefonie, Streaming, Online-Multiplayer

Hard real-time systems: Airbags, ABS, Ventilautomatiken

Aufgabe 2.5: Prozessorausnutzung

(Tafelübung)

Die Prozessorausnutzung ρ sei als Quotient aus der minimal erforderlichen und der tatsächlich benötigten Zeit zur Ausführung anstehender Prozesse definiert. Dabei soll die Laufzeit eines Prozesses T Zeiteinheiten betragen und ein Prozesswechsel S Zeiteinheiten kosten (es gilt: $S \ll T$).

- Geben Sie eine alternative Formel zur Berechnung der Prozessorausnutzung für das Round-Robin-Verfahren unter Verwendung der Zeitscheibenlänge τ und der Prozesszahl n an.
- Berechnen Sie anhand der Formel aus a) die Grenzwerte für folgende Fälle:
 - $\tau \rightarrow 0$,
 - $\tau = S$ und
 - $\tau \rightarrow \infty$.
- Stellen Sie die Abhängigkeit von Effizienz und Zeitscheibenlänge grafisch dar.

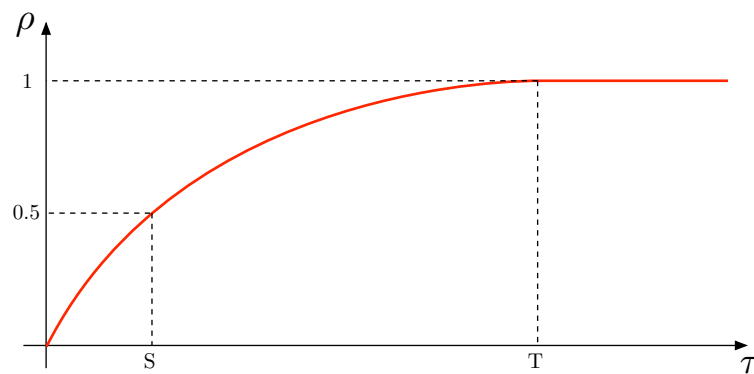
Lösung:

- Kurzform der Herleitung:

Wie bereits in der Aufgabenstellung beschrieben, ist die Prozessorausnutzung der Quotient aus der minimal erforderlichen und der tatsächlich benötigten Zeit: $\rho = \frac{z_{\min}}{z_{\text{ges}}}$. Für diese beiden Werte gilt: $z_{\min} = n \cdot T$ und $z_{\text{ges}} = (n \cdot \lceil \frac{T}{\tau} \rceil - 1) \cdot S + n \cdot T$. Daraus resultiert die folgende Formel:
$$\rho = \frac{n \cdot T}{(n \cdot \lceil \frac{T}{\tau} \rceil - 1) \cdot S + n \cdot T}$$

- Für verschiedene Zeitscheiben ergeben sich andere Reaktionszeiten und Prozessorauslastungen. Kleinere Zeitscheiben ermöglichen dem System scheinbar schnellere Reaktionen, da viele Prozesse nacheinander kleine Aufgaben erledigen können. Bei großen Zeitscheiben ist die Auslastung höher und die mittlere Antwortzeit von Prozessen geringer, aber Desktops oder Multimediaanwendungen können stockend bzw. verzögert erscheinen.
 - für $\tau \rightarrow 0$ gilt: $\rho \approx 0$, da nur noch zwischen den Prozessen gewechselt, aber nicht mehr gerechnet wird.
 - für $\tau = S$ gilt: $\rho = \frac{n \cdot T}{(n \cdot \lceil \frac{T}{\tau} \rceil - 1) \cdot S + n \cdot T} = \frac{T}{\lceil \frac{T}{\tau} \rceil \cdot S - \frac{S}{n} + T} \approx \frac{T}{2 \cdot T} = \frac{1}{2}$ (Die -1 kommt daher, weil nach dem letzten Prozess kein Wechsel mehr notwendig ist; bei hinreichend großem n und kleinem S kann man den Term vernachlässigen.)
 - für $\tau \rightarrow \infty$ gilt: Auslastung $\rho = \frac{n \cdot T}{(n-1) \cdot S + n \cdot T} \approx \frac{T}{S+T}$ (im Grunde FCFS und da $S \ll T$: $\rho \approx 1$)

c) Aus a) und b) lässt sich folgender Graph schlussfolgern:



Aufgabe 2.6: Scheduling-Handsimulation (Selbststudium)

Es werden in einem Ein-Prozessor-System Prozesse wie in Abbildung 3 beschrieben gestartet:

Prozess	A	B	C	D	E
Ankunftszeitpunkt	0	2	3	6	8
Dauer	7	4	3	2	4

Tabelle 3: Prozesse eines Systems mit einer CPU und einem Thread

a) Simulieren Sie folgende Scheduling-Verfahren für die Prozesse aus Abbildung 3:

- SRTN,
- HRRN,
- MLF mit $\tau_i = 2^i$ ($i = 0, 1, \dots$)

Bei gleichen Prioritäten verwenden Sie das FCFS-Verfahren.

Geben Sie für *jeden* Zeitpunkt den Inhalt der Warteschlange und den Prozess auf der CPU an.

Die Lösung soll in Form der dargestellten Tabelle abgegeben werden, wobei anzumerken ist, dass für Multilevel-Feedback mehrere Warteschlangen benötigt werden:

Zeit	0	1	2	3	...	18	19
CPU	A
Warteschlange	

b) Berechnen Sie für *jedes* der in a) verwendete Verfahren

- die Warte- und Antwortzeit *jedes* Prozesses sowie
- die mittlere Warte- und Antwortzeit des gesamten Systems.

Lösung:

SRTN:

Zeit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
CPU	A7	A6	B4	B3	B2	B1	D2	D1	C3	C2	C1	E4	E3	E2	E1	A5	A4	A3	A2	A1
WS			A5	C3	C3	C3	C3	C3	E4	E4	E4	A5	A5	A5	A5					
				A5	A5	A5	A5	A5	A5	A5	A5									

HRRN:

Zeit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
CPU	A	A	A	A	A	A	A	C	C	C	B	B	B	B	D	D	E	E	E	E
WS			B	B	B	B	C	B	B	B	D	D	D	D	E	E				
				C	C	C	B	D	D	D	E	E	E	E						
							D		E	E										

MLF:

Zeit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
CPU	A7	A6	A5	B4	C3	B3	B2	D2	E4	C2	C1	D1	E3	E2	A4	A3	A2	A1	B1	E1
Level 0			B4	C3			D2													
Level 1					B3	C2	C2	C2	D1	D1	E3	D1	E3	E3						
Level 2				A4	A4	A4	A4	A4	B1	A4	B1	A4	B1	A4	B1	B1	E1	B1	E1	E1

	SRTN		HRRN		MLF	
	Wartezeit	Antwortzeit	Wartezeit	Antwortzeit	Wartezeit	Antwortzeit
A	13	20	0	7	11	18
B	0	4	8	12	13	17
C	5	8	4	7	5	8
D	0	2	8	10	4	6
E	3	7	8	12	8	12
System	4.2	8.2	5.6	9.6	8.2	12.2