



SoSe 20

Systemprogrammierung

## Aufgabenblatt 3

Abgabe von

Michelle Döring

Jannik Heinze

Iryna Novytska

Charlotte Fritsch

## Aufgabe 3.4: Scheduling-Theorie

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

Das highest response ratio next Scheduling Verfahren ist eine vorhersagebasierte CPU-Zuteilungsstrategie, die das mögliche Verhungern von CPU-lastigen Prozessen vermeidet. Dabei wird das Altern(aging), d.h. die Wartezeit von Prozessen mit berücksichtigt. Ausgewählt wird immer der Prozess mit dem größten  $rr = \frac{\text{Wartezeit} + \text{Bedienzeit}}{\text{Bedienzeit}}$ .  
[Quelle: Vorlesung 7, S.24, <https://ess.cs.tu-dortmund.de/Teaching/SS2013/BS/Downloads/04-Scheduling.pdf>, letzter Abruf: 09.06.2020, 20:40]

Jeder Prozesswechsel kostet zusätzlich Zeit. Da das HRRN-Verfahren ohne Verdrängung arbeitet, hat man mit den größten **Durchsatz**, weil Prozesse einfach immer nur damit beschäftigt sind bearbeitet zu werden und nicht immer neu berechnet werden muss, welcher Prozess als nächstes dran ist.

Auch Prozesse die sehr lange brauchen haben bei highest response ratio next eine Chance irgendwann auch gescheduled zu werden. Demnach ist dieses Verfahren besonders 'fair'.  
[Quelle: Tafelübung zu Blatt 3]

Damit bildet das HRRN-Verfahren einen Kompromiss zwischen dem Schedulingziel Fairness und dem Schedulingziel Durchsatz/Effizienz.

### b) Erklären sie das Phänomen der Prioritätsinvertierung. 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 2).

Prioritätsinvertierung ist ein Problem beim Prioritätsscheduling, bei dem ein Prozess verhungert, obwohl er der dringlichste ist.

In unserem Beispiel wird  $M$  gestartet und wird aufgrund der höheren Priorität den aktuell laufenden Prozess  $L$  verdrängen. Prozess  $L$  kann dadurch jedoch nicht das Betriebsmittel  $A$  freigeben. Demnach bleibt  $M$  der höchst-priorisierte lauffähige Prozess und der eigentlich dringlichste Prozess  $H$  muss bis zur Fertigstellung von  $M$  und dann  $L$  warten.  
[Quelle: Vorlesung 7, S.21]

### c) Was ist ein möglicher Lösungsansatz für dieses Problem? Erläutern sie diesen kurz.

Ein möglicher Lösungsansatz für dieses Problem wäre die sogenannte "Prioritätsvererbung" (priority inheritance). Dabei bekommt der das Betriebsmittel blockierende Prozess die Priorität des dringlichsten Prozesses, solange er das Betriebsmittel blockiert, auf das der dringlichste Prozess wartet.

In unserem Fall bekommt  $L$  also die Priorität von  $H$ , damit er die Dringlichkeit des Prozesses  $H$  'simulieren' kann und sicherstellt, dass  $H$  in der Prioritätsliste an der richti-

gen Stelle ausgeführt werden kann. Dadurch wird nun  $L$  nicht von dem neu gestarteten  $M$  verdrängt, sondern zuerst zuende geführt, kann dann das Betriebsmittel freigeben und  $H$  wird lauffähig. Als nächstes wird  $H$  ausgeführt und anschließend  $M$ .

**d) Was sind die Unterschiede zwischen Online- und Offline-Scheduling? Gehen sie dabei auch auf die benötigten Voraussetzungen für beide ein.**

Beim Offline-Scheduling wird die Zuteilung einmalig berechnet und zwar vor der Ausführungszeit. Zur Laufzeit steht der Ablaufplan bereits komplett fest, d.h. es müssen keine Entscheidungen mehr getroffen werden. Damit dies möglich ist, müssen bereits sämtliche notwendigen Informationen über die Prozesse zur Verfügung stehen. Also sowohl die Prozesse, als auch zukünftige Ankünfte.

Beim Online-Scheduling hingegen wird vollständig zur Laufzeit agiert. Dabei müssen ggf. dynamisch auf Grundlage der aktuellen Situation Entscheidungen getroffen werden. Es sind jeweils nur die aktuellen Prozesse bekannt und bei Ankunft neuer Prozesse muss neu entschieden werden.

[Quelle: Vorlesung 7, S.7, <https://docplayer.org/18729348-B-worin-besteht-der-unterschied-zwischen-online-und-offline-scheduling.html>, letzter Abruf: 09.06.2020, 20:40]

**e) Was sind die Unterschiede und Gemeinsamkeiten von Hard- und Soft-real-Systems? Nennen sie jeweils ein Beispiel.**

Ein wesentlicher Unterschied liegt in der Tolerierung von Verletzungen der Sollzeitpunkte. Während die Hard-real-time-Systems strikte Einhaltung der Sollzeitpunkte fordern, so sind in schwachen Echtzeitsystemen (Soft-real-time-Systems) die Verletzungen tolerierbar, führen jedoch zur Qualitätsverlust (Quelle: Folien Kapitel 3, S.32). Ein weiterer Unterschied ist in der Antwortzeit (Response Time) der beiden Systeme. Bei Hard-real-time-Systems liegt diese in Größenordnung von Millisekunden, während Soft-real-time-Systems eine längere Antwortzeit brauchen. Die strikten Echtzeitsysteme werden im gegenatz zu der anderen Variante in Systemen mit hohen Anforderungen an Sicherheit verwendet. Weitere Unterschiede liegen in der Fehlerdetektion, Datenintegrität, Größe der Daten-Files etc.

[Quelle: [https://users.ece.cmu.edu/~koopman/des.s99/real\\_time/](https://users.ece.cmu.edu/~koopman/des.s99/real_time/), letzter Abruf: 09.06.2020, 20:40]

Hard-real-Systems sind vorhersagbar wohingegen Soft-real-Systems effizient und besser anpassbar an dynamische Daten sind.

[Quelle: <https://www.springer.com/de/book/9780387237015>, letzter Abruf: 09.06.2020, 20:40]

Die Gemeinsamkeiten liegen darin, dass sie beide real-time-Systems sind, also ist bei beiden die Systemzeit mit der externen Zeitreferenz synchronisiert. Alle Messungen erfolgen im Bezug auf diese externe Referenz.

[Quelle: <https://www.springer.com/de/book/9780387237015>, S.2]

## Aufgabe 3.5: Scheduling-Handsimulation

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

### SRTN

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

### 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
Warteschlange		B	B,C	B,C	B,C	B,C	B,C,D	B,D	B,D,E	B,D,E	D,E	D,E	D,E	D,E	E	E	E			

MLF mit  $\tau = 2^i (i = 0, 1, \dots)$  (Ohne Verdrängung)

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	B	C	B	B	D	E	C	C	D	E	E	A	A	A	A	B	E
LVL0 ( $\tau = 2^0 = 1$ )			B	C			D													
LVL1 ( $\tau = 2^1 = 2$ )					B	C	C	C	C,D	D,E	D,E	E								
LVL2 ( $\tau = 2^2 = 4$ )				A	A	A	A	A,B	A,B	A,B	A,B	A,B	A,B	A,B	B,E	B,E	B,E	B,E	E	

b) Berechnen Sie für jedes der in a) verwendeten Verfahren die Warte- und Antwortzeit jedes Prozesses sowie die mittlere Warte- und Antwortzeit des gesamten Systems.

### SRTN

Prozess	Wartezeit	Antwortzeit
A	13	20
B	0	4
C	5	8
D	0	2
E	3	7
mittlr. Warte und Antwortzeit	4,2	8,2

### HRRN

Prozess	Wartezeit	Antwortzeit
A	0	7
B	8	12
C	4	7
D	8	10
E	8	12
mittlr. Warte und Antwortzeit	5,6	9,6

**MLF**

Prozess	Wartezeit	Antwortzeit
A	11	18
B	13	17
C	5	8
D	4	6
E	8	12
mittlr. Warte und Antwortzeit	8,2	12,2