



Betriebssysteme und Sicherheit, WS 2019/20

## 4. Aufgabenblatt – Quantitative Methoden

Geplante Bearbeitungszeit: eine Woche

**Aufgabe 4.1** Gegeben sei die Präzedenzrelation  $\{(A,C), (B,C), (C,D), (E,D), (E,F)\}$  zwischen sechs Jobs  $A, \dots, F$ . Die Jobs erfordern folgende Bedienungszeiten (in einer bestimmten Zeiteinheit):

$A$	$B$	$C$	$D$	$E$	$F$
2	3	3	2	5	4

Zur Bearbeitung stehen zwei gleichartige Prozessoren zur Verfügung. Die Jobs sind nicht unterbrechbar.

(a) Zeichnen Sie den Präzedenzgraphen (einschließlich der Bedienungszeiten) als Vorgangspfeilnetz. Beachten Sie dabei die folgenden Regeln, die für Vorgangspfeilnetze gelten:

- Es darf nur *einen* Anfangsknoten und *einen* Endknoten geben.
- Zwischen zwei Knoten darf es nur *eine* Kante (ohne weitere Zwischenknoten) geben.
- Kanten sollten sich nicht kreuzen.
- Es sollten so wenig Scheinvorgänge als möglich verwendet werden.
- Alle Pfeile (alle Kanten des Graphen) müssen von links nach rechts gerichtet sein.

Überlegen Sie sich auch, welches die Motivationen für diese Regeln sind!

(b) Geben Sie die Ablaufpläne (Gantt-Diagramme) gemäß FIFO (dies sei die in der Tabelle stehende Reihenfolge), SPT und LPT an. Berechnen Sie in allen Fällen die relevanten Bewertungsgrößen! Erläutern Sie (unter Einbeziehung dieser Ergebnisse) die Bedeutung der einzelnen Zuteilungsstrategien. Welcher Ablaufplan ist optimal?

(c) Versuchen Sie, einen Ablaufplan zu konstruieren, der gleichzeitig zu maximalem Durchsatz und zu minimaler mittlerer Jobverweilzeit führt. Vergleichen Sie die Ergebnisse mit den in (b) ermittelten Werten.

(d) Unter welchen Modifikationen der Voraussetzungen lassen sich prinzipiell weitere Verbesserungen erreichen? Inwieweit ist dies im konkreten Fall möglich?

(e) Inwieweit ändern sich die Ergebnisse, wenn nur ein Prozessor zur Verfügung steht?

**Aufgabe 4.2** In einem Betriebssystem erfolge das Scheduling nach dem Round-Robin-Verfahren mit der Zeitscheibenlänge  $Q$ .  $n$  Jobs, von denen jeder den (einigen) Prozessor für  $T = k \cdot Q$  Zeiteinheiten ( $k \in \mathbb{N}^+$ ) beansprucht, werden gleichzeitig in die Warteschlange bereiter Jobs eingereiht. Berechnen Sie die Wartezeit für jeden Job und die mittlere Wartezeit für die Menge der  $n$  Jobs. Was folgt daraus für den Fall des sog. Prozessor-Sharings (Zeitquant der Größe 0)?

Hinweis: Betrachten Sie zunächst bei  $n = 4$  und  $T = 10$  die drei Fälle  $Q = 10, Q = 5, Q = 1$ .

**Aufgabe 4.3** Geben Sie einen Überblick über das Scheduling in Echtzeitsystemen (Aufgaben und Begriffe – Modellannahmen – wichtige Scheduling-Verfahren und deren Eigenschaften – Kriterien für die Zulassung einer Task-Menge).

**Aufgabe 4.4** In einem Ein-Prozessor-Echtzeitsystem sind zwei periodische Tasks  $T_1, T_2$  einzuplanen mit folgenden Werten ( $p_i$  Periodenlänge,  $t_i$  Bearbeitungszeit, Periodenende = Zeitschranke):

$$p_1 = 5 \text{ s}, t_1 = 2 \text{ s}$$

$$p_2 = 3 \text{ s}, t_2 = 1 \text{ s}$$

Zwischen den Tasks bestehen keine Abhängigkeiten; sie sind an beliebiger Stelle unterbrechbar. Die Tasks sollen auf möglichst einfache Weise (geringer Overhead) eingeplant werden.

- Untersuchen Sie, inwieweit dies möglich ist.
- Nach einiger Zeit ist eine weitere Task  $T_3$  einzuplanen mit  $p_3 = 5 \text{ s}$ ,  $t_3 = 1 \text{ s}$ . Diskutieren Sie die entstandene Situation.
- Nachträglich stellt sich heraus, dass für  $T_1$  ein Prozessorbedarf von 2,1 s erforderlich ist. Welche Konsequenzen hat dies für Zulassung und Einplanung der drei Tasks?

## Klausuraufgabe I

Ein um die Erde kreisender Satellit bestimmt seine Position mittels dreier verschiedener Lagesensoren (Sternensensor A, Magnetometer B, Gyroskop C). Zur genauen Lageregelung müssen diese periodisch ausgewertet werden, um die Antriebe entsprechend zu steuern. Die notwendigen Perioden  $p$  und Ausführungszeiten  $e$  der Auswertungsalgorithmen sind in nebenstehender Tabelle zusammengefasst. Die relative Deadline  $d$  ist gleich der Periodenlänge ( $d = p$ ). Alle Tasks sind voneinander unabhängig und laufende Jobs zu beliebiger Zeit unterbrechbar.

	A	B	C
$p$	9	6	3
$e$	3	2	1

- Ist die Lageregelung auf einem Prozessor mittels dynamischer Prioritäten möglich, falls der Scheduling-Overhead vernachlässigbar ist? Wenn nein, begründen Sie. Falls ja, schlagen Sie einen Algorithmus für die Einplanung vor.
- Während der Entwicklung der Software wird festgestellt, dass der Scheduling-Overhead für das Einplanen eines neuen Jobs in einem System mit dynamischen Prioritäten 0,2 Zeiteinheiten beträgt. Ist die Taskmenge weiterhin auf einem Prozessor einplanbar?
- Existiert eine statische Prioritätszuordnung mit welcher die Taskmenge auf einem Prozessor eingeplant werden kann? Falls ja, geben Sie diese Zuordnung an! Ansonsten begründen Sie Ihre Aussage!  
Hinweis: Bei statischen Prioritäten ist der Scheduling-Overhead vernachlässigbar.

Als Lösung schlägt der System-Designer vor, den Sternensensor (A) durch einen Sonnensensor (A') zu ersetzen. Dessen Auswertungsalgorithmus hat eine Periode von 4 und eine Ausführungszeit von 1 Zeiteinheiten. Das System besteht damit jetzt aus den nebenstehenden Tasks.

	A'	B	C
$p$	4	6	3
$e$	1	2	1

- Ist dieses System mittels statischer Prioritäten einplanbar? Falls ja, nennen Sie eine Prioritätszuteilung! Ansonsten begründen Sie Ihre Aussage!  
Hinweis: Bei statischen Prioritäten ist der Scheduling-Overhead vernachlässigbar.
- Wäre das System mittels dynamischer Prioritäten einplanbar, wenn hier weiterhin von einem Overhead für den Jobwechsel von 0,2 Zeiteinheiten ausgegangen wird?

## Klausuraufgabe II

In einem Echtzeitsystem ist eine Menge von drei periodischen Tasks so einzuplanen, dass deren Jobs in jeder Periode erfolgreich beendet werden. Das Periodenende entspricht der Deadline. Die Parameter der Tasks beschreiben jeweils die Periode  $p$  und die konstante Bearbeitungszeit  $e$  der Jobs.

$$T_1: p_1 = 3, e_1 = 1$$

$$T_2: p_2 = 12, e_2 = 2$$

$$T_3: p_3 = 4, e_3 = 1$$

Alle Tasks starten zum Zeitpunkt  $t = 0$  und können an beliebiger Stelle unterbrochen werden. Der Scheduling-Overhead werde vernachlässigt und es stehe genau ein Prozessor zur Verfügung.

- Zunächst werden den Tasks statisch Prioritäten zugeteilt, so dass  $T_1$  die höchste,  $T_2$  eine mittlere und  $T_3$  die niedrigste Priorität erhält. Weisen Sie nach, dass die gegebene Task-Menge unter den genannten Bedingungen nicht einplanbar ist!
- Ist eine Einplanung dieser Task-Menge mit statischen Prioritäten überhaupt möglich? Wenn ja, geben Sie eine entsprechende Prioritätszuteilung an!
- Die Task-Menge soll mit dynamischen Prioritäten eingeplant werden. Schlagen Sie ein passendes Einplanungsverfahren vor und geben Sie an, welcher Task im gewählten Verfahren zum Zeitpunkt  $t = 3,5$  die höchste Priorität aufweist.
- Weiterhin soll die Ausführungszeit  $e_3$  der Task  $T_3$  erhöht werden. Bis zu welchem Maximalwert von  $e_3$  ist eine Einplanung mit dynamischen Prioritäten möglich?