

Betriebssysteme im Wintersemester 2018/2019

Übungsblatt 10

Abgabetermin: 14.01.2019, 18:00 Uhr

Besprechung: Besprechung der T-Aufgaben in den Tutorien vom 07. – 11. Januar 2019
Besprechung der H-Aufgaben in den Tutorien vom 14. – 18. Januar 2019

Aufgabe 44: (T) Leser-/Schreiberproblem mit Petrinetz

(– Pkt.)

Beim Leser-/Schreiberproblem operieren n Leserprozesse und m Schreiberprozesse auf ein und derselben Datei. Um Inkonsistenzen der Dateiinhalte zu vermeiden, müssen die folgenden beiden Bedingungen erfüllt sein:

- Mehrere Leserprozesse dürfen zur gleichen Zeit auf die Datei zugreifen.
- Ein Schreiberprozess darf nur dann auf die Datei zugreifen, wenn gerade kein anderer Leser- oder Schreiberprozess auf die Datei zugreift.

Bearbeiten Sie unter Berücksichtigung dieser beiden Bedingungen die folgenden Aufgaben

- a. Modellieren Sie das Leser-/Schreiberproblem als ein Petrinetz. Gehen Sie dabei von der folgenden Situation aus:
- Es gibt drei Leserprozesse und drei Schreiberprozesse, die auf ein und dieselbe Datei lesend bzw. schreibend zugreifen wollen. Es handelt sich dabei um disjunkte Prozesse.
 - Es können maximal zwei Leserprozesse gleichzeitig auf die Datei zugreifen.

Hinweise:

- Leserprozesse (Schreiberprozesse) können entweder auf die Datei lesend (schreibend) zugreifen, oder warten auf ihren Lesezugriff (Schreibzugriff). Überlegen Sie sich basierend darauf zunächst, welche Zustände Ihr Petrinetz modellieren muss.
 - Modellieren Sie die Leser- und Schreiberprozesse als Marken!
 - Zählen Sie unter Verwendung einer separaten Stelle in Ihrem Petrinetz mit, wieviele Leserprozesse bzw. Schreiberprozesse noch Zugriff auf die Datei erhalten dürfen.
 - Verwenden Sie an geeigneter Stelle gewichtete Transitionen.
- b. Skizzieren Sie den Erreichbarkeitsgraphen für das in Aufgabe a) modellierte Petrinetz
- c. Handelt es sich bei Ihrer Modellierung aus Aufgabe a) um ein *fares* Petrinetz, d.h. können hier Prozesse verhungern? Begründen Sie Ihre Antwort! Für den Fall, dass Ihre Lösung kein *fares* Petrinetz darstellt: Wie könnte man ein Verhungern unterbinden?

Aufgabe 45: (T) Prozessfortschrittsdiagramm

(– Pkt.)

Bearbeiten Sie die folgenden Aufgaben.

- a. Gegeben seien zwei Prozesse P und Q, die auf einem Uniprozessorsystem ausgeführt werden sollen. Der Prozess P benötigt 9 und der Prozess Q 10 Zeiteinheiten für seine Ausführung. Es stehen die Betriebsmittel BM 1–6 zur Verfügung, die von den Prozessen während ihrer Ausführung benötigt werden.

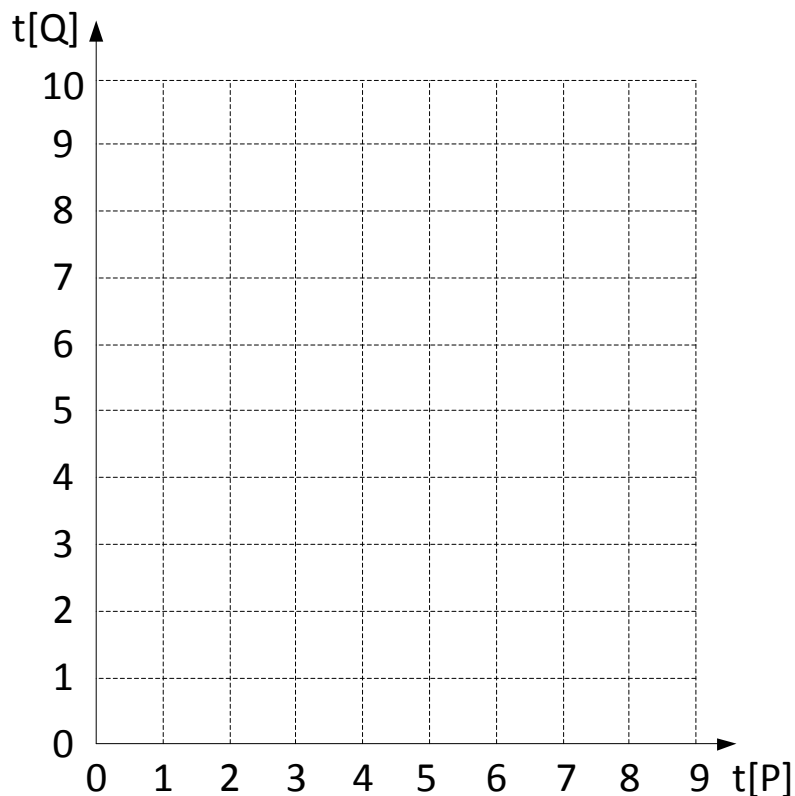
Q benötigt:

- BM1 im Zeitraum]1; 3[,
- BM2 im Zeitraum]1; 2[,
- BM3 im Zeitraum]6; 7[,
- BM4 im Zeitraum]1; 2[,
- BM5 im Zeitraum]5; 7[und
- BM6 im Zeitraum]4; 6[.

P benötigt:

- BM1 im Zeitraum]2; 3[,
- BM2 im Zeitraum]2; 4[,
- BM3 im Zeitraum]4; 6[,
- BM4 im Zeitraum]5; 7[,
- BM5 im Zeitraum]5; 8[und
- BM6 im Zeitraum]7; 8[.

Skizzieren Sie das Prozessfortschrittsdiagramm für die oben beschriebenen Anforderungen, indem Sie die benötigten Betriebsmittel entsprechend ihrer zeitlichen Verwendung durch die beiden Prozesse P und Q korrekt einzeichnen. Gehen Sie davon aus, dass der Scheduler die Prozesse P und Q zu beliebigen Zeitpunkten aktivieren bzw. suspendieren kann. Gehen Sie zudem davon aus, dass ein Kontextwechsel zwischen P und Q keinerlei Zeit in Anspruch nimmt. Tragen Sie Ihre Lösung in die folgende Vorlage ein:



- b. Kennzeichnen Sie **deutlich** alle unmöglichen und unsicheren Bereiche im Diagramm aus Teilaufgabe a).
- c. Zeichnen Sie alle *prinzipiell* unterschiedlichen Ausführungspfade (d.h., dass Prozess P und Q anders ge-scheduled werden und dementsprechend die Betriebsmittel in unterschiedlicher Reihenfolge nutzen) in das Diagramm aus Teilaufgabe a) ein, so dass die Prozesse P und Q terminieren.
- d. Bezogen auf Teilaufgabe a): Wieviele *prinzipiell* unterschiedliche Ausführungspfade gibt es (d.h., dass Prozess P und Q anders ge-scheduled werden und dementsprechend die Betriebsmittel in unterschiedlicher Reihenfolge nutzen), die in einem Deadlock enden? Geben Sie für jeden solchen Ablauf ein Beispiel an und beschreiben sie dabei, wann und wie lange Prozess P bzw. Q aktiviert bzw. suspendiert werden muss, um in eine Deadlock-Situation zu gelangen.

Aufgabe 46: (H) Algorithmus von Peterson

(15 Pkt.)

Druckaufträge werden vom Betriebssystem in einer (FIFO-)Warteschlange W verwaltet. Die Warteschlange verwaltet selbst lediglich eine Liste von Zeigern, die auf den Speicherbereich verweisen, an dem die zu druckenden Daten liegen. Die Variable `next` enthält den Index der nächsten freien Position in der Warteschlange.

Gegeben seien zwei Prozesse P_1 und P_2 , die jeweils eine Datei drucken möchten. Die folgende Tabelle illustriert die Programmausschnitte, die die Prozesse P_1 bzw. P_2 dazu jeweils ausführen.

Prozess P_1

```

1 ...
2 W[next] = pointer_file1;
3 next = next + 1;
4 ...

```

Prozess P_2

```

1 ...
2 W[next] = pointer_file2;
3 next = next + 1;
4 ...

```

- a. Welches Problem kann auftreten, wenn P_1 und P_2 im Mehrprogrammbetrieb parallel ausgeführt werden? Modellieren Sie einen Ablauf, der dieses Problem illustriert. Verwenden Sie dazu folgende Darstellung:

aktiver Prozess	ausgeführte Codezeile	Inhalt von W	Wert von <code>next</code>	Kommentar
...

Stellen Sie den Inhalt von W als Liste der Form $[ptr_1, ptr_2, \dots]$ dar.

- b. Synchronisieren Sie die Prozesse mit dem Algorithmus von Peterson. Geben Sie dazu (in Analogie zu den Code-Ausschnitten der Angabe) die Codeausschnitte der Prozesse P_1 und P_2 an.
- c. Welchen erheblichen Nachteil hat der Peterson-Ansatz?

Aufgabe 47: (H) Einfachauswahlaufgabe: Prozesskoordination

(5 Pkt.)

Für jede der folgenden Fragen ist eine korrekte Antwort auszuwählen („1 aus n“). Nennen Sie dazu in Ihrer Abgabe explizit die jeweils ausgewählte Antwortnummer ((i), (ii), (iii) oder (iv)). Eine korrekte Antwort ergibt jeweils einen Punkt. Mehrfache Antworten oder eine falsche Antwort werden mit 0 Punkten bewertet.

a) Wie bezeichnet man die Eigenschaft einer korrekten Lösung des wechselseitigen Ausschlusses, die besagt, dass sich zu jedem Zeitpunkt nur ein Prozess im kritischen Bereich befinden darf?			
(i) Bounded Waiting	(ii) Progress	(iii) Mutual Exclusion	(iv) Circular Wait
b) Ein Computer habe vier Bandlaufwerke und n Prozesse, von denen jeder zwei Bandlaufwerke gleichzeitig für seine Ausführung benötigt. Bei einer Anfrage bekommt ein Prozess ein beliebiges freies Bandlaufwerk zugewiesen. Für einen Prozess ist es irrelevant welche Bandlaufwerke er verwendet, solange es zwei sind. Nachdem er die zwei Bandlaufwerke erhalten hat, terminiert er nach endlicher Zeit. Für welchen Wert von n besteht die Möglichkeit eines Deadlocks?			
(i) 1	(ii) 2	(iii) 3	(iv) 4
c) Für eine korrekte Lösung des wechselseitigen Ausschlusses müssen die drei Bedingungen Mutual Exclusion, Progress, und Bounded Waiting erfüllt sein. Welche Bedingung(en) erfüllt der Algorithmus von Decker (erster Ansatz) nicht?			
(i) Progress	(ii) Mutual Exclusion	(iii) Bounded Waiting	(iv) alle drei
d) Für eine korrekte Lösung des wechselseitigen Ausschlusses müssen die drei Bedingungen Mutual Exclusion, Progress, und Bounded Waiting erfüllt sein. Was trifft auf den Algorithmus von Peterson zu?			
(i) Er erfüllt keine der Bedingungen.	(ii) Er erfüllt alle Bedingungen.	(iii) Er erfüllt nur die Mutual Exclusion Bedingung.	(iv) Er erfüllt nur die Progress Bedingung.
e) Was ist keine der 3 atomaren Operationen, mit denen ein Semaphore S verändert werden kann?			
(i) <code>block(S)</code> oder auch <code>B(S)</code>	(ii) <code>wait(S)</code> oder auch <code>P(S)</code>	(iii) <code>init(S, Anfangswert)</code>	(iv) <code>signal(S)</code> oder auch <code>V(S)</code>