

Betriebssysteme im Wintersemester 2019/2020

Übungsblatt 8

Abgabetermin: 16.12.2019, 18:00 Uhr

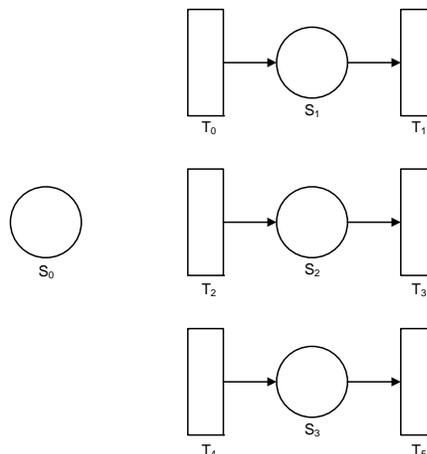
Besprechung: Besprechung der T-Aufgaben in den Tutorien vom 09. – 13. Dezember 2019
 Besprechung der H-Aufgaben in den Tutorien vom 16. – 20. Dezember 2019

Aufgabe 38: (T) Petri-Netze: Multiprocessing

(– Pkt.)

In dieser Aufgabe betrachten wir ein Zwei-Prozessor-System mit drei rechenbereiten Prozessen P_1 , P_2 und P_3 . Gehen Sie zunächst von folgenden Bedingungen aus:

- Auf einer CPU kann immer nur genau ein Prozess gleichzeitig rechnen.
 - Ein Prozess kann auf einer oder aber auch gleichzeitig auf beiden CPUs abgearbeitet werden
- a. Welches Thread-Konzept (KLT oder ULT) muss ein Betriebssystem umsetzen, damit die zweite der oben genannten Bedingungen erfüllt ist? Begründen Sie Ihre Antwort!
- b. Im folgenden sollen das Zwei-Prozessor-System und die drei Prozesse mit einem Petri-Netz modelliert werden. Die Stellen dieses Petri-Netzes können als Zustände interpretiert werden. Dabei entspricht S_0 dem Zustand *“Ein bzw. zwei Prozessoren stehen zum Rechnen zur Verfügung”* und die Stelle S_1 (S_2 , S_3) entspricht dem Zustand *“ P_1 (P_2 , P_3) rechnet auf einem bzw. zwei der Prozessoren”*. Ergänzen Sie nun das nachfolgend angedeutete Petri-Netz um eine minimale Anzahl an weiteren Marken, Stellen, Transitionen und Kanten (insofern jeweils erforderlich), so dass zu jedem Zeitpunkt entweder genau ein Prozess beide Prozessoren nutzen kann, oder zwei Prozesse genau je einen Prozessor nutzen können.



- c. Geben Sie zu dem erarbeiteten Petri-Netz den Erreichbarkeitsgraphen an.
- d. Ist das System Deadlock-frei, teilweise verklemmt oder verklemmt? Begründen Sie Ihre Antwort mit Hilfe des eben skizzierten Erreichbarkeitsgraphen!
- e. Um die Architektur des Systems besser debuggen zu können, sollen Prozesse so beschränkt werden, dass sie immer nur auf genau einer CPU zur selben Zeit ausgeführt werden können. Dazu muss also sichergestellt werden, dass auf den beiden Prozessoren zu jedem Zeitpunkt nur unterschiedliche Prozesse rechnen können.

Man kann diese Einschränkung sehr einfach modellieren, indem man die Kapazität der Stellen S_1 , S_2 und S_3 auf 1 limitiert. Erweitern Sie das Petri-Netz aus Aufgabe b) so, dass Ihre Modellierung den selben Effekt erzielt, dabei aber keine Einschränkung der Kapazität einer Stelle voraussetzt. Lösen Sie das Problem durch Verwendung einer minimalen Anzahl an Stellen, Marken, Kanten und Transitionen.
- f. Geben Sie zu dem erweiterten Petri-Netz aus Aufgabe e) den Erreichbarkeitsgraphen an.

Aufgabe 39: (T) Leser-/Schreiberproblem mit Petri-Netz

(– 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 Petri-Netz. 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 Petri-Netz modellieren muss.
 - Modellieren Sie die Leser- und Schreiberprozesse als Marken!
 - Zählen Sie unter Verwendung einer separaten Stelle in Ihrem Petri-Netz 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 Petri-Netz
 - c. Handelt es sich bei Ihrer Modellierung aus Aufgabe a) um ein *fares* Petri-Netz, d.h. können hier Prozesse verhungern? Begründen Sie Ihre Antwort! Für den Fall, dass Ihre Lösung kein *fares* Petri-Netz darstellt: Wie könnte man ein Verhungern unterbinden?

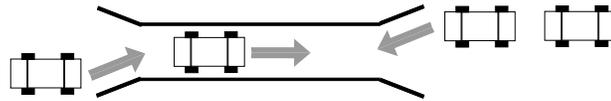
Aufgabe 40: (H) Modellierung eines Petri-Netzes

(13 Pkt.)

In dieser Aufgabe sollen Sie mit Hilfe eines Petri-Netzes eine Einbahnbrücke und vier Fahrzeuge modellieren. Dabei sollen folgende Bedingungen gelten.

- Über die Einbahnbrücke kann zu einem Zeitpunkt immer nur genau ein Fahrzeug fahren.
- Ein Fahrzeug darf auf der Einbahnbrücke nicht die Fahrtrichtung wechseln.
- Ein Fahrzeug, das die Brücke überquert hat, reiht sich unter den Fahrzeugen ein, die die Brücke in der anderen Richtung überqueren wollen.
- Unter den Fahrzeugen, die die Brücke überqueren wollen existiert keine Ordnung, d.h. es wird stets zufällig ein Fahrzeug ausgewählt, das als nächstes die Brücke überquert.

Die folgende Abbildung veranschaulicht eine Situation, in der sich gerade ein Fahrzeug auf der Einbahnbrücke befindet, ein Fahrzeug darauf wartet, die Brücke von links nach rechts zu überqueren und zwei Fahrzeuge darauf warten, die Brücke von rechts nach links zu überqueren.



Bearbeiten Sie unter den oben geschilderten Voraussetzungen nun folgende Aufgaben.

- a. Modellieren Sie die Einbahnbrücke und die vier Fahrzeuge als ein Petri-Netz. Gehen Sie dabei davon aus, dass sich zu Beginn je zwei Fahrzeuge auf jeder Seite der Brücke befinden. Auf der Brücke selbst befindet sich zu Beginn kein Fahrzeug. Beschriften Sie alle Stellen und Transitionen Ihres Petri-Netzes! **Hinweis:** Modellieren Sie die Situation in der sich ein Fahrzeug von links nach rechts bzw. von rechts nach links bewegt jeweils als eine eigene Stelle.
- b. Skizzieren Sie den Erreichbarkeitsgraphen für das in Aufgabe a) modellierte Petri-Netz. Notieren Sie an jeder Kante des Erreichbarkeitsgraphen die Transition, die feuern muss, um den Übergang zwischen zwei Zuständen Ihres Petri-Netzes zu erwirken.
- c. Handelt es sich bei Ihrer Modellierung aus Aufgabe a) um ein faires Petri-Netz? Begründen Sie Ihre Antwort!

Aufgabe 41: (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 $next$ | 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 42: (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) Welche Art von Prozessen macht die Synchronisation der Prozesse untereinander erforderlich? | | | |
| (i) unabhängige Prozesse | (ii) sequenzielle Prozesse | (iii) nebenläufige Prozesse | (iv) parallele aber unabhängige Prozesse |
| b) Wie bezeichnet man einen Speicher, der in Form eines eindimensionalen Arrays unter Verwendung der Modulo-Funktion realisiert wird? | | | |
| (i) Ringpuffer | (ii) Linearpuffer | (iii) Wechsellpuffer | (iv) Sparpuffer |
| c) 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 |
| d) 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 |
| e) 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. |