

Beispiellösung - Übungsblatt 5

Erstellt von Daniel Bub

Aufgabe 1.

Recherchieren Sie das ‘Sleeping-Barber-Problem’. Welche Problematik steckt hinter diesem Problem?

Lösung 1.

Situation:

Ein Friseur bietet seine Dienste in seinem Friseursalon, der aus einem Arbeitszimmer mit einem Stuhl und einem Wartezimmer mit n Warteplätzen besteht, an. Nachdem er einem Kunden die Haare geschnitten hat, holt er den nächsten Kunden aus dem Wartezimmer ab, um diesem die Haare zu machen. Sollte allerdings kein neuer Kunde warten, geht er zurück und legt sich auf seinem Friseurstuhl schlafen. Kommt ein neuer Kunde ins Wartezimmer und sieht, dass keine Person vor ihm wartet, geht er direkt in das Arbeitszimmer des Friseurs. Schläft der Friseur, weckt er diesen, ansonsten geht er zurück ins Wartezimmer und nimmt Platz.

Problematik:

Die Zeit, die der Friseur zum Überprüfen des Wartezimmers benötigt, und die Zeit, die der Kunde braucht, um den ‘Status’ des Friseurs zu überprüfen, können unterschiedlich lang sein.

Ablauf-Beispiel:

Der Friseur beendet die Arbeit an einem Kunden und geht das Wartezimmer überprüfen. Gleichzeitig trifft ein neuer Kunde ein, der erkennt, dass das Wartezimmer leer ist. Er macht sich auf den Weg, um im Arbeitszimmer nachzusehen (**Friseur und neuer Kunde treffen sich nicht!!!**).

Der Friseur sieht, dass sich niemand im Wartezimmer befindet und geht zurück, um sich auf seinem Stuhl schlafen zu legen. Der neue Kunde bemerkt ebenfalls, dass der Friseur nicht in seinem Zimmer ist, geht zurück ins Wartezimmer und nimmt Platz. Noch ein neuer Kunde trifft ein, sieht, dass Wartezimmer bereits besetzt und wartet ebenfalls. Somit kann der Friseur von keinem Kunde mehr geweckt werden \Rightarrow **Deadlock**.

Aufgabe 2.

Erläutern Sie grob die Lösung des Problems.

Lösung 2.

Eine mögliche Lösung für dieses Problem könnte sein, dass maximal eine Person gleichzeitig den Raum wechseln darf. Programmiertechnisch mit einem **Mutex** möglich. Zusätzlich können die Zustände der einzelnen Personen bzw. die Anzahl der wartenden Kunden mit **Semaphoren** verwaltet werden.

Aufgabe 3.

Warum gibt es für Prozesse keinen direkten Übergang vom Zustand ‘blockiert’ in den Zustand ‘rechnend’?

Lösung 3.

Blockierte Prozesse warten auf die Bereitstellung von Daten, etwa Nutzereingaben. Sobald diese Daten vorliegen, und der Prozess theoretisch wieder rechnen könnte, wird er jedoch erst als “rechenbereit” betrachtet. Das bedeutet, dass der Scheduler diesem Prozess Rechenzeit zuweisen kann. Erst wenn diese Zuweisung erfolgt ist, befindet sich der Prozess wieder im Zustand “rechnend”.

Aufgabe 4.

In zwei Tagen ist Prüfung! Sie bereiten sich eifrig auf die Prüfung vor und möchten schnell noch *den* Tanenbaum und ein Linux-Grundlagenbuch ausleihen. Wie kann ein Deadlock entstehen?

Lösung 4.

Folgendes Beispielszenario wäre denkbar:

1. Sie leihen das Linux-Grundlagenbuch aus und bekommen den Hinweis, dass der Tanenbaum bereits ausgeliehen ist.
2. Die andere Person, die sich *den* Tanenbaum bereits ausgeliehen hat, benötigt zum weiteren Selbststudium das von Ihnen ausgeliehene Linux-Grundlagenbuch.
3. Weder Sie, noch die andere Person, geben das jeweils ausgeliehene Buch zurück, bevor sie nicht das andere Buch erhalten haben.
4. Sie können aber ohne das Betriebssysteme-Buch von Tanenbaum nicht weiterarbeiten und warten bis es in der Bücherei wieder verfügbar ist.

⇒ **Eine Deadlock-Situation ist eingetreten.**

Aufgabe 5.

Oje, Sie fielen durch die Prüfung! (Hoffentlich nicht!) Welche Konsequenz ziehen Sie, um beim nächsten Versuch (auf legalem Wege) an die notwendigen Bücher zu gelangen?

Lösung 5.

Den Tanenbaum und das Linux-Grundlagenbuch rechtzeitig reservieren und beide gleichzeitig wieder zurück geben.