
Concurrent Programming

<http://www-pu.informatik.uni-tuebingen.de/cp-2006>

Blatt 2

Abgabe: 16.5.2006

1. [5 Punkte] Schreibe ein Scheme-Programm, das zwei Threads startet, von denen einer den Buchstaben **a** und der andere den Buchstaben **b** wiederholt ausdrückt. In der Ausgabe sollen die beiden Buchstaben immer genau abwechselnd vorkommen.
2. [5 Punkte] Die Beschreibung der Mutex-Locks in der Vorlesung hat einige Sonderfälle noch nicht abgedeckt. So ist zum Beispiel noch nicht klar, welcher Thread das Mutex-Lock erhält, wenn mehrere Threads auf dessen Freigabe warten. Welches Verhalten würdest Du in diesem Fall für sinnvoll befinden? Begründe Deine Antwort!

Finde noch mindestens zwei weitere Fälle, die durch die Beschreibung in der Vorlesung noch nicht abgedeckt sind und beschreibe, welches Verhalten Du jeweils für sinnvoll erachtest. Begründe auch hier deine Antworten.
3. [10 Punkte] Viele Mittel zur Synchronisation von Threads lassen sich durch Mutex-Locks implementieren, so auch Semaphoren. Programmiere **make-semaphore**, **semaphore-post** und **semaphore-wait** mit Mutex-Locks. Implementiere die Semaphoren als einen neuen SRFI-9-Recordtyp.
4. [10 Punkte] In der Vorlesung wurde eine Lösung für das Producer/Consumer-Problem vorgestellt, bei der Zugriff auf den Puffer über die Variable **waiting** und die Semaphore **wait** synchronisiert wurde. Programmiere eine Lösung für das Producer/Consumer-Problem, die ohne **waiting** auskommt. Synchronisiere den Zugriff auf den Puffer stattdessen mit zwei Semaphoren **full** und **empty**.