Lehrstuhl für Programmiersprachen und Übersetzerkonstruktion
CAU Kiel, Informatik und Prakt. Mathematik
Frank Huch

24118 Kiel
Christan-Albrechts-Platz 4
Tel.: 0431 / 880-7277

1. Übung „Nebenläufige und verteilte Programmierung“

Abgabe am 29. Oktober in der Vorlesung

Aufgabe 1 **6 Punkte**

In der Vorlesung wurde das Erzeuger-/Verbraucherproblem mit Hilfe von Semaphoren gelöst. Hierbei gingen wir von einem unbeschränkten Puffer aus.

a) In der Implementierung mit synchronisiertem Zugriff auf den Puffer führt der Erzeuger zwei V-Operationen und der Verbraucher zwei P-Operationen hintereinander aus. Was geschieht, wenn man die Reihenfolge der Operationen vertauscht?

b) Die Synchronisation auf den Puffer ist nicht immer nötig. Überlegen Sie, wie Sie die Synchronisation auf den Puffer verbessern können (Einschränken des wechselseitigen Ausschlusses).

c) Welche Änderungen müssen Sie vornehmen, wenn Sie einen endlichen Puffer verwenden?

Aufgabe 2 **4 Punkte**

Die in der Vorlesung vorgestellte Implementierung der dinierenden Philosophen zur Vermeidung eines Deadlocks (Zurücklegen des Stabs, falls der andere bereits belgt ist) ist noch nicht korrekt.

a) Geben Sie eine Ausführung mit zwei Philosophen an, bei welcher dennoch ein Deadlock entstehen kann.

b) Korrigieren Sie das Programm durch Hinzunahme weiterer Semaphoren zur zusätzlichen Synchronisation.

Aufgabe 3 **2 Punkte**

Schreiben Sie ein sequentielles Java-Programm, welches die Zahlen von 1 bis n auf dem Bildschirm ausgibt. n sei hierbei ein Parameter mit welchem das Java-Programm (`A3`) gestartet wird:

```
java A3 n
```