



Nebenläufiges Programmieren

Sommersemester 2008

Serie 12

25.06.2008

Ausgabetermin: 25.06.2008

Abgabe: 04.07.2008 (11:00)

Aufgabe 1 (4 Punkte) Der *Linda's tuple space* kann als Server implementiert werden. Wollen Anwendungsprogramme dabei eine *Linda primitive* ausführen (z.B. OUT, IN oder EVAL), so interagieren sie mit dem Server.

Entwickle Pseudo-Code für einen solchen Serverprozess, der einen *Lina's tuple space* mit den sechs Grundfunktionen (*Linda primitives*, i.e., OUT, IN, RD, INP, RDP, EVAL) implementiert. Gebe ebenfalls (passenden) Pseudo-Code für ein Anwendungsprogramm an, dass den zuvor entwickelten Server und damit die implementierten Grundfunktionen verwendet.

Die Kommunikation zwischen Anwendungsprogramm und dem Server soll auf asynchronem *message passing* basieren.

Kommentiere Deinen Code. Verwende keine komplexen Datentypen/-strukturen. Falls spezielle Operationen nötig sind, die vorausgesetzt werden müssen, sind diese ausreichend detailliert bei der Beschreibung der Gesamtidee zu spezifizieren und zu erklären (z.B. Parametertypen, Rückgabewerte, Funktionsweise,...).

Aufgabe 2 (4 Punkte) Betrachte die folgende Spezifikation eines Programms, welches das Minimum einer Menge von Integer-Zahlen ermittelt:

Gegeben sei ein *array* von n Prozessen $\text{Min}[1:n]$. Jeder Prozess hat initial einen (beliebigen aber festen) Integer-Startwert. Die Prozesse interagieren fortlaufend miteinander und versuchen dabei einander stets das von ihnen bisher wahrgenommene Minimum aller bis dahin empfangener Startwerte und dem eigenen Startwert weiterzureichen. Sobald ein Prozess seinen solchen (partiellen) Minimalwert weitergereicht hat, beendet er sich. Schliesslich bleibt ein einziger Prozess übrig, der damit das Minimum aller Startwerte kennt.

- (a) Entwickle ein MPD-Programm, dass eine korrekte Implementierung der zuvor beschriebenen Spezifikation darstellt und die RPC-Grundfunktionen aus [Andrews, 8.1] zur Kommunikation verwendet.
- (b) Entwickle ein MPD-Programm, dass eine korrekte Implementierung der zuvor beschriebenen Spezifikation darstellt und die Rendezvous-Grundfunktionen aus [Andrews, 8.2] zur Kommunikation verwendet.

Kommentiere Deine Lösung ausreichend gründlich und beschreibe Deine Lösungsansätze und Vorgehensweise. Füge Deinen Abgaben außerdem jeweils ebenfalls kommentierte Probeläufe bei. Argumentiere insbesondere über die korrekte Implementierung der Spezifikation.
Zur Erinnerung: Die Abgabe von Programmieraufgaben erfolgt immer (auch) per E-Mail.