Christian-Albrechts-Universität zu Kiel
Institut für Informatik
Prof. Dr. W.-P. de Roever
Jens Schönborn, Christian Motika und Tim Fenten

Nebenläufiges Programmieren

Sommersemester 2008 — **Serie 13 (EST)** — 30.06.2008

Ausgabetermin: 30.06.2008

Abgabe: 11.07.2008 (11:00)

Diese Serie soll in Einzelarbeit gelöst werden! Auch Abschreiben lassen führt zur Disqualifikation. Der Gedankenaustausch zu den Aufgaben(stellungen) ist (beschränkt) nur in den Übungsstunden gestattet.

Jede Aufgabe soll auf einem neuen Zettel begonnen werden. Verseht bitte alle Zettel die ihr abgebt mit eurem Namen.

Aufgabe 1 (4 Punkte) ***Multiple Primitives Notation*** **-** Die Abbildung 7.15 aus [Andrews] zeigt wie der Algorithmus *Sieb des Eratosthenes* zum Identifizieren von Primzahlen mit Hilfe von Filter-Prozessen realisiert werden kann. Dabei wurde das *CSP synchronous message passing* verwendet.
In der Abbildung 7.16 aus [Andrews] werden Primzahlen mit Hilfe des *bag of tasks*-Ansatzes unter Verwendung von *C-Linda* gefunden.

(a) Schreibe das Programm aus Abbildung 7.15 [Andrews] unter Beibehaltung des dortigen Algorithmus so um, dass der Algorithmus die *multiple primitives notation*, wie in Abschnitt 8.3 [Andrews] definiert verwendet.

(b) Schreibe das Programm aus Abbildung 7.16 [Andrews] unter Beibehaltung des dortigen Algorithmus so um, dass der Algorithmus die *multiple primitives notation*, wie in Abschnitt 8.3 [Andrews] definiert verwendet.

Pseudocode genügt. Erkläre Deine Vorgehensweise, erläutere und kommentiere den Programmcode. Gehen insbesondere auf die gemachten Änderungen ein und mache deutlich, dass das neue Programm (bezogen auf den Algorithmus) sich entsprechend gleich verhält.

Aufgabe 2 (2 Punkte) ***Readers/Writers*** **-** Der *readers/writers*-Algorithmus in Abbildung 8.13 [Andrews] bevorzugt Leser. Ändere den Algorithmus und damit speziell das `IN`-Statement derart ab, dass der Algorithmus Lesern und Schreibern abwechselnd den Vorzug gibt. Pseudocode genügt. Erläutere Deine Änderungen und argumentiere mit Hilfe der (aus [Andrews] ersichtlichen) Semantik für das `IN`-Statement, weshalb Deine Lösung korrekt ist.

Aufgabe 3 (6 Punkte) ***Stable Marriage Problem*** **-** Seien `Man` und `Woman` zwei *array*s von `n` Prozessen die `n` Männer und Frauen repräsentieren. Jeder Mann bewertet alle Frauen in einer (hier: beliebigen) Rangordnung von `1` bis `n` und jede Frau bewertet alle Männer ebenfalls von `1` bis `n`.
Eine *Paarung* ist eine vollständige Einteilung der Männer und Frauen in `n` *Paare*. Ein *Paar* ist eine 1-zu-1-Beziehung zwischen genau einem Mann und genau einer Frau.
Paare haben die Eigenschaft *stabil* zu sein, genau dann wenn für zwei *Paare* von Männern und Frauen (m_1, w_1) und (m_2, w_2) beide der folgenden Bedingungen erfüllt sind:

1. m_1 bewertet w_1 höher als w_2 oder w_2 bewertet m_2 höher als m_1
2. m_2 bewertet w_2 höher als w_1 oder w_1 bewertet m_1 höher als m_2

Anders ausgedrückt ist ein *Paar* als *instabil* anzusehen, wenn ein Mann und eine Frau sich beide einander gegenüber ihrem aktuellen (durch die *Paar*-Beziehung vorgegeben) Partner bevorzugen würden.
Eine Lösung für das *stable marriage problem* ist eine *Paarung*, in der alle `n` *Paar*-Beziehungen *stabil* sind.
Schreibe ein `mpd`-Programm mit Hilfe der *multiple primitives notation* aus Abschnitt 8.3 [Andrews], um das *stable marriage problem* für eine beliebige Anzahl von `n` Männern und Frauen mit beliebigen Bewertungen für einander zu lösen.
Erkläre Deine Vorgehensweise, erläutere und kommentiere den Programmcode. Mache an Deinem Code ebenfalls deutlich, dass er die gegebene Spezifikation in korrekter Weise erfüllt.

Aufgabe 4 (3 Punkte) ***Time Server*** **-**Schreibe den Time Server Prozess in Abbildung 8.7. derart um, dass die **delay** Operation ein Zeitintervall zum Schlafen wie in Abbildung 8.1 spezifiert anstatt einen gegebenen Aufwachzeitpunkt. Benutze ausschliesslich Rendenzvous-Primitiven wie in Section 8.2 beschrieben. (Tip: Es werden eine oder mehrere zusätzliche Operationen benötigt und die Clients werden nicht einfach delay aufrufen können.) Pseudocode genügt. Kommentiere den Code und erkäre die Funktionsweise. Gebe auch den relevenaten Code der Clients an.