



Nebenläufige Programmierung

Wintersemester 2004/05

Serie 2

26. Oktober 2004

Thema:

Ausgabetermin: 26. Oktober 2004

Abgabe: 1. November 2004 (12:00)

Aufgabe 1 (6 Punkte) Schreibe ein iteratives paralleles Programm um das folgende Problem zu lösen. Erzeuge eine $n \times n$ Matrix, wobei n als Kommandozeilenparameter übergeben werden soll. Initialisiere die Matrix an jeder Position zufällig mit 0 oder 1 (mit gleicher Wahrscheinlichkeit).

Führe folgende Berechnungen aus:

1. Berechne die Anzahl der Vorkommen 1 in der Matrix.
2. Finde heraus ob die Matrix symmetrisch ist (d.h. ob $M[i, j] = M[j, i]$ für alle i, j ist).

Benutze dazu p Prozesse, wobei p ebenfalls als Kommandozeilenparameter zu übergeben ist, jeder Prozess soll einen Teil der Matrix berechnen. Der Prozess darf seinen Teil der Matrix selbst initialisieren. Man darf dabei annehmen dass n ein Vielfaches von p ist.

Nimm nicht an, dass Zuweisungen an gemeinsame Variablen atomar ausgeführt werden.

Das Programm soll ferner die erzeugte Matrix in eine Datei schreiben.

Aufgabe 2 (2 Punkte) Aufgaben bzgl. ‘sequential consistency’:

1. Gegeben sei folgende Sequenz von M_{serial} :

$$W_2(5, y) \quad W_3(6, y) \quad W_1(4, y) \quad R_3(4, y)$$

Geben Sie alle Sequenzen von SCM (bestehen nur aus read and write Befehlen) die mit obiger M_{serial} -Sequenz harmonieren können (analog zu der Tafel rechts oben auf Seite 58 des Handouts).

2. Geben Sie eine Sequenz in M_{distr} an, deren Projection auf read and write obige Sequenz liefert. Welche Werte können am Ende aus Variable y gelesen werden?
3. Sei M'_{distr} wie M_{distr} bis auf das die Bedingung ‘ \wedge no \star -ed entries in In_i ’ in Fig. 4 entfernt ist. Zeigen sie, dass M'_{distr} nicht ‘sequentially consistent’ ist. Beschreiben Sie in Worten für was die ‘ \star -ed entries’ stehen.
4. Bilden Sie M''_{distr} indem sie Fig. 4 entsprechend abändern, so dass kein Cache größer als m benutzt wird und man mit möglichst maximalen Cache arbeitet.