

# Verlässliche verteilte Systeme – Aufgabenblatt 2

**Bearbeitung zum 14.12.2009**

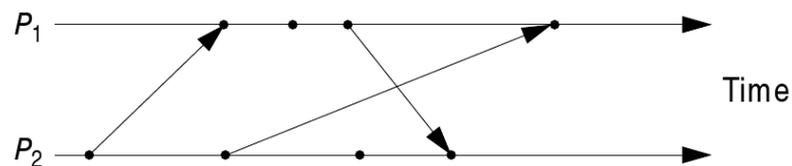
## 1. Aufgabe

Damit aktive Replikation funktionieren kann, ist es im Allgemeinen erforderlich, dass alle Operationen auf jedem Replikat in derselben Reihenfolge erfolgen. Ist diese Anordnung immer nötig?

## 2. Aufgabe

Um Multicasting mit einer totalen Ordnung der Nachrichten in einem verteilten System zu implementieren, existieren unterschiedliche Möglichkeiten. Nennen Sie drei alternative Ansätze und vergleichen Sie die Lösungen miteinander!

## 3. Aufgabe



Die Abbildung zeigt die Ereignisse in zwei Prozessen P1 und P2. Nachrichtenübertragungen werden durch Pfeile gekennzeichnet. Die globalen Zustände des Systems können als  $S(i, j)$  notiert werden, wobei  $S(i, j)$  der Zustand nach  $i$  Ereignissen in P1 und  $j$  Ereignissen in P2 ist. Der Ausgangszustand ist  $S(0, 0)$ .

- a) Welche Zustände sind strikt konsistent?
- b) Die folgende Tabelle enthält alle möglichen globalen Zustände. Streichen Sie die inkonsistenten Zustände!

$S(0, 0)$	$S(0, 1)$	$S(0, 2)$	$S(0, 3)$	$S(0, 4)$
$S(1, 0)$	$S(1, 1)$	$S(1, 2)$	$S(1, 3)$	$S(1, 4)$
$S(2, 0)$	$S(2, 1)$	$S(2, 2)$	$S(2, 3)$	$S(2, 4)$
$S(3, 0)$	$S(3, 1)$	$S(3, 2)$	$S(3, 3)$	$S(3, 4)$
$S(4, 0)$	$S(4, 1)$	$S(4, 2)$	$S(4, 3)$	$S(4, 4)$

## 4. Aufgabe

Der Schnappschuss-Algorithmus von Chandy/Lamport erlaubt es, einen konsistenten Zustand aller Prozesse und Kommunikationskanäle eines verteilten Systems aufzuzeichnen.

- a) Können Konsistenzprobleme auftreten, wenn mehrere Prozesse den Algorithmus zur gleichen Zeit initiieren? Wie kann man diese Probleme beheben?
- b) Eine Voraussetzung für den Algorithmus ist, dass alle Kommunikationskanäle nach FIFO arbeiten. Was kann passieren, wenn diese Bedingung nicht erfüllt ist?

## 5. Aufgabe

**Behauptung:** der View Synchronous View-Change Algorithmus kann dazu verwendet werden, eine totale Ordnung aller Ereignisse herzustellen, indem man periodisch einen View-Change durchführt.

Begründen oder widerlegen Sie diese Behauptung!