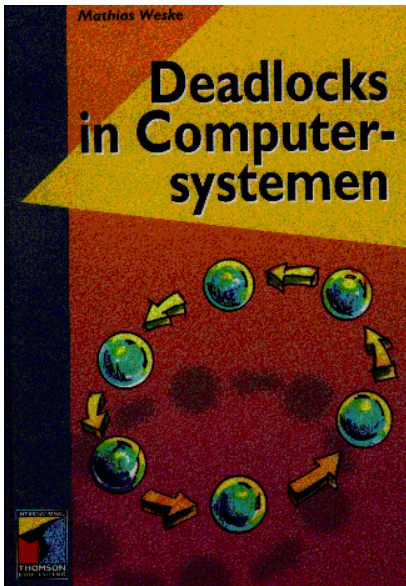


# Deadlocks in Computer Systemen

Mathias Weske, International Thomson Publishing, ISBN 3-929821-11-7, 440 Seiten, öS 616,-

Norbert Bartos



Das vorgenannte Buch wurde in vier Teilen organisiert:

- 1) Grundlagen: Allgemeines, Deadlock-Problem, Charakterisierung von Deadlocks, Deadlockbehandlung, Kosten, Modellierung (Graphen), Beschreibungstechnik
- 2) Deadlocks in Betriebssystemen
- 3) Deadlocks in Datenbanksystemen
- 4) Deadlocks in Rechnernetzen

Die Kapitel 2) bis 4) besitzen folgenden identischen Aufbau: Grundlagen, Deadlockentstehung, Verhinderung,

Vermeidung, Erkennung, Auflösung, Diskussion von Algorithmen

Durch den weitgehend identischen Aufbau der tragenden Abschnitte ergeben sich naturgemäß einige Wiederholungen, was aus didaktischen Gründen durchaus positiv zu sehen ist. Aufgrund der reichlich vorhandenen Beispiele ist das Buch zum Selbststudium vorzüglich geeignet. Ein über das normale Maß an Grundkenntnissen hinausgehendes Wissen ist nicht als Voraussetzung nötig. Allerdings ist es für den HTL- und FH-Bereich eher nicht allgemein verwendbar, da es das Deadlock-Problem sehr breit analysiert, was aus Zeitgründen in den genannten Schulformen aber nicht möglich ist. Wohl kann es aber als Basis für Seminararbeiten dienen.

Ein kurzer Abschnitt aus dem Buch möge den Stil erläutern. Hierbei geht es um die Modellierung von Deadlocks mittels eines speziellen Wartegraphen, des „Ressourcen-Allokationsgraphen“. Das im Text angesprochene **Beispiel 1.8** ist der immer wiederkehrende Fall einer Kreuzung, auf der vier Fahrzeuge gleichzeitig erscheinen und diese geradlinig passieren wollen. Dabei ist **F** ein Fahrzeug und **R** sein Einfahrtsbereich in die Kreuzung, wie aus dem folgenden Originalbild aus Seite 28 zu entnehmen ist.

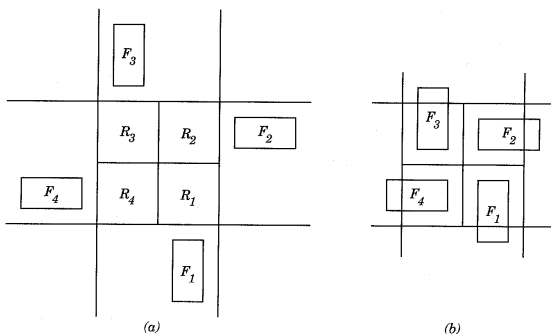


Abb. 1.8: Kreuzungs-Beispiel: (a) formalisierte Darstellung, (b) Deadlocksituation

Ein Deadlock führt dazu, daß das gesamte System ins Stocken gerät. In Computersystemen treten an Stelle der Kreuzungsbereiche die Ressourcen (Betriebsmittel wie Drucker, Platte, Datei), die Fahrzeuge entsprechen den Prozessen (Tasks, Transaktionen). Ein Deadlock bewirkt einen kompletten Systemstillstand, der nur durch rigorose Maßnahmen, wie Abbrechen von Tasks, behoben werden kann. Die Betriebssystemhersteller sind daher eifrig bemüht, Verfahren zur Deadlockverhinderung oder -vermeidung, bzw. zur Erkennung und Beseitigung zu entwickeln und zu implementieren.

Der im nachfolgenden Originalkapitel 2.3.2 (Seite 59-60) angeführte Graph kann mit einem geeigneten Algorithmus auf Deadlocksituationen analysiert werden. Im Falle eines Deadlocks enthält der Graph mindestens einen Zyklus.

### 2.3.2 Ressource-Allokationsgraphen

Das in Abschnitt 2.2 entwickelte Systemmodell verwaltet Informationen über die Betriebsmittelsituation von Prozessen und die sich daraus ergebenden Wartbeziehungen, in Allokations-, Anforderungs- und Wartemengen. Diese Informationen können durch gerichtete Graphen veranschaulicht werden.

**Definition 2.8** Ein *Ressource-Allokationsgraph (RAG)* ist ein bipartiter gerichteter Graph  $G = (V, E)$ , dessen Knotenmenge  $V$  partitioniert ist in eine Menge  $\mathcal{P}$  von Prozessen und eine Menge  $\mathcal{R}$  von Betriebsmitteln und dessen Kantenmenge  $E$  partitioniert ist in eine Menge  $E_A \subseteq \mathcal{R} \times \mathcal{P}$  von *Allokationskanten* und eine Menge  $E_R \subseteq \mathcal{P} \times \mathcal{R}$  von *Anforderungskanten* (oder *Request-Kanten*) mit:

$$E_A := \{(R_j, P_i) \mid R_j \in AS_i\}$$

$$E_R := \{(P_i, R_j) \mid R_j \in RS_i\}$$

◊

In *Ressource-Allokationsgraphen*<sup>3</sup> werden Prozeßknoten durch Kreise und Ressource-Knoten durch Rechtecke dargestellt. Diese Graphen enthalten zunächst keine Informationen über Wartbeziehungen zwischen Prozessen, da sie nicht auf Wartemengen basieren, sondern auf Allokations- und Anforderungsmengen. Sie enthalten lediglich Informationen darüber, welche Betriebsmittel an welche Prozesse zugewiesen sind und welche Prozesse auf die Zuweisung welcher Betriebsmittel warten. Allerdings können diese Informationen – wie im vorhergehenden Abschnitt diskutiert – dazu verwendet werden, Wartbeziehungen zwischen Prozessen zu ermitteln.

<sup>3</sup>Da RAGs nicht nur Allokationskanten, sondern auch Anforderungskanten enthalten, müßten sie genauer mit „Ressource Allokations- und Anforderungsgraphen“ bezeichnet werden.

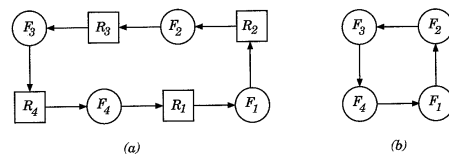


Abb. 2.4: Kreuzungs-Deadlock: (a) Ressource-Allokationsgraph, (b) Wartegraph

**Beispiel 2.18** Die Abbildung 2.4(a) enthält den *Ressource-Allokationsgraphen* des Kreuzungs-Deadlock aus Abbildung 1.8(b), dessen Allokations- und Anforderungsmengen in Beispiel 2.15 ermittelt wurden.

□