



**Übung zur Vorlesung
„Einsatz und Realisierung von Datenbanksystemen“
im Sommersemester 2007**

Richard Kuntschke (richard.kuntschke@in.tum.de)

Lösungen zu Blatt 1

Aufgabe 1

Die Datenobjekte A und D liegen auf einer Seite P_A auf dem Hintergrundspeicher. Die Pufferseiten-Ersetzungsstrategien sind wie folgt definiert:

- steal*: Jede nicht fixierte Seite ist prinzipiell ein Kandidat für die Ersetzung, falls neue Seiten eingelagert werden müssen.
- \neg *steal*: Seiten, die Änderungen von noch aktiven Transaktionen enthalten, dürfen nicht ausgelagert werden.
- force*: Beim **commit** einer Transaktion werden **alle** von ihr modifizierten Seiten in die Datenbasis übertragen.
- \neg *force*: Von einer Transaktion modifizierte Seiten müssen nach dem **commit** der Transaktion nicht notwendigerweise in die Datenbasis eingebracht werden.

Wir zeigen im Folgenden einen Beispielschedule zweier Transaktionen T_1 und T_2 , der eine Kombination der Strategien *force* und \neg *steal* unmöglich macht.

Beispielschedule

Schritt	T_1	T_2
1.	BOT	
2.		BOT
3.	read(A)	
4.		read(D)
5.		write(D)
6.	write(A)	
7.	commit	⋮

Wegen der *force* Strategie beim Einbringen von Änderungen und der \neg *steal* Strategie bei der Seitenersetzung entsteht bei Schritt 7 ein Konflikt:

- Wegen *force* müssten die Änderungen der Transaktion T_1 auf der Seite P_A auf den Hintergrundspeicher propagiert werden.
- Wegen \neg *steal* dürfen aber die Änderungen der noch aktiven Transaktion T_2 auf der Seite P_A nicht auf den Hintergrundspeicher gelangen.

Selbst eine Verzögerung des **commit** von T_1 würde das Problem nicht lösen, wenn T_2 als Nächstes versucht, eine Sperre auf A (die ja noch von T_1 gehalten wird) zu erlangen.

- ⇒ Die bekannten Mechanismen für den Wiederanlauf sind anwendbar, d.h. auch in diesem Fall ist eine Recovery möglich.
- Das Log muss evtl. genutzt werden, um nach einem Verlust der Datenbasis aus einem Archivzustand der Datenbasis die aktuelle Version wieder herzuleiten.
- ⇒ Hierzu muss das Log komplett vorliegen.

Aufgabe 5

Die Einträge des Log-Archives müssen bis zum Eintrag mit LSN $MinLSN$ reichen, wobei $MinLSN$ die LSN des ältesten Eintrags einer Transaktion ist, die beim Schreiben des Sicherungspunktes noch nicht abgeschlossen war.

Wiederherstellen des DB-Zustandes

1. Einspielen des DB-Archivs.
2. Analyse setzt im Log zum Zeitpunkt des Schreibens des Archivs auf.
 - Auslesen der Liste der Transaktionen, die zum Zeitpunkt des Schreibens des Archivs (= aktionskonsistenter Sicherungspunkt) aktiv waren.
 - Ermittlung der Winner und Loser, unter Beachtung der obigen Liste.
3. Ausführen des *Redo* ab Archivierungszeitpunkt.
4. Ausführen des *Undo* (zurück bis maximal $MinLSN$).