Ceng 352 Written Homework 3 – Solutions

Online Part

WHW3 Question 1 – Automatically graded

WHW3 Question 2- Solution:

1st question:

Allowed by 2PL. A schedule that obeys 2PL is:

T1: lock-X(A), T1: R(A), T1: W(A), T1: unlock(A), T2: lock-X(A), T2: R(A), T2: W(A),

T2: lock-X(B), T2: R(B), T2: W(B), T2: unlock(A), T2:unlock(B), T2: Commit, T1: Abort

2nd question:

Not allowed by Strict 2PL. This is because Strict 2PL requires T1 and T2 to hold the locks until completion, but T2 cannot proceed once T1 holds an exclusive lock on A.

PDF part

Q1)

(a) Yes, it is guaranteed because the transactions follow strict 2PL.

(b) Yes, deadlock is possible. If T1 starts first and acquires X(A) and then T2 starts and acquires X(B) then deadlock will occur as T1 is blocked on request of X(B) and T2 is blocked on request of X(A). In the wait-die scheme (only) the higher priority transaction is allowed to wait, so T2 would roll back.

(c) Not possible, since transactions follow Strict 2PL, and all strict schedules are cascadeless.

(d) Yes, it is guaranteed because the transactions follow (non strict) 2PL.

(e) Not possible, because the locks are acquired in an ordered manner.

(f) Yes, it is possible. In the following schedule, the abort of T2 cascades and requires the abort of T1:

T2:X(A) T2:X(B) T2:R(B) T2:R(A) T2:W(A) T2:U(A) T1:X(A) T1:R(A)

T1:W(A) T2:Abort

Q2)

a) i.

Operation	A			B			C		
	RTS	WTS	C	RTS	WTS	C	RTS	WTS	C
R1(A)	1								
R2(B)				2					
R3(A)	3								
W1(A): Rejected Abort T1	3			2					
R2(C)							2		
W3(B)				2	3	F			
W2(C)						F	2	2	F
R2(A)	3					F			F
W3(C)						F	2	3	F
Commit T3	3			2	3	T	2	3	T
W2(B): Ignore Thomas' rule				2	3	T			T
Commit T2	3			2	3	T	2	3	T

b) ii.

Operation	A			B			C		
	RTS	WTS	C	RTS	WTS	C	RTS	WTS	C
R1(A)	2								
R2(B)				3					
R3(A)	2								
W1(A)	2	2	F						
R2(C)							3		
W3(B): rejected Abort T3									
W2(C)							3	3	F
Commit T1	2	2	T	3			3	3	F
R2(A)	3								
W2(B)					3	F			
Commit T2	3	2	T	3	3	T	3	3	T

Q3)

a) State of the System just Before Crash

Transaction Table:

Tid	LastLSN	status
T3	14	Active
T4	13	Commit

Dirty Page Table:

pageId	recLSN
P1	2
P2	3

Pages in Memory

PageId	Value	pageLSN
P1	A2 B2	14
P2	C D1	10

Master record: 5 (LSN of begin checkpoint)

Disk: (P1 was flushed to the disk)

Page	Values	pageLSN
P1	A1 B	1
P2	C D	

LOG:

LSN	Tid	prevLSN	Type	pageID	Log entry	undoNextLSN
1	**T1**	**-**	**Update**	**P1**	**Write A(A->A1)**	**-**
2	**T2**	**-**	**Update**	**P1**	**Write B (B->B1)**	**-**
3	**T2**	**2**	**Update**	**P2**	**Write C (C->C1)**	**-**
4	**T2**	**3**	**Abort**			
5	**Begin checkpoint**					
6	**TT:[(T1,1,active)(T2,4,abort)],DPT:[(P1 2),(P2 3)] End checkpoint**					
7	**T2**	**4**	**CLR**		**Undo T2 LSN 3**	**2**
8	**T2**	**7**	**CLR**		**Undo T2 LSN 2**	**-**
9	**T2**	**8**	**End**			
10	**T3**	**-**	**Update**	**P2**	**Write D (D->D1)**	**-**
11	**T1**	**1**	**Commit**			
12	T1	11	End			
13	T4	-	Update	P1	Write A (A->A2)	
14	T3	10	Update	P1	Write B (B->B2)	
15	T4	13	Commit			

CRASH

The LOG was last flushed to the disk when T1 committed.
Only boldface lines will be found on disk when crash recovery begins. The log will be read starting from LSN 5.

The LOG found after the crash:

LSN	Tid	prevLSN	Type	pageID	Log entry	undoNextLSN
1	**T1**	**-**	**Update**	**P1**	**Write A(A->A1)**	**-**
2	**T2**	**-**	**Update**	**P1**	**Write B (B->B1)**	**-**
3	**T2**	**2**	**Update**	**P2**	**Write C (C->C1)**	**-**
4	**T2**	**3**	**Abort**			
5	**Begin checkpoint**					
6	**TT:[(T1,1,active)(T2,4,abort)],DPT:[(P1 2),(P2 3)]** **End checkpoint**					
7	**T2**	**4**	**CLR**		**Undo T2 LSN 3**	**2**
8	**T2**	**7**	**CLR**		**Undo T2 LSN 2**	**-**
9	**T2**	**8**	**End**			
10	**T3**	**-**	**Update**	**P2**	**Write D (D->D1)**	**-**
11	**T1**	**1**	**Commit**			

---

b) Transaction Table and Dirty Page table right after Analysis

Transaction Table

Tid	LastLSN	status
T3	10	Active
T1	11	Commit

Dirty page Table:

pageId	recLSN
P1	2
P2	3

c) Redo starts at LSN 2, because that is the smallest LSN in the Dirty Page Table.

All log records 2,3,7,8 will be redone. After Redo phase the pages in the memory are:

Pages in Memory

PageId	Value	pageLSN
P1	A1 B	8
P2	C D1	10

d) The contents of the LOG at the end of the Undo phase:

Transaction table contains T1 and T3. T1 committed so End record will be written to the log. T3 was an active transaction, its effects must be undone.

toUNDO ={10}

The following records are appended to the LOG

LSN	Tid	prevLSN	Type	pageID	Log entry	undoNextLSN
1	**T1**	**-**	**Update**	**P1**	**Write A(A->A1)**	**-**
2	**T2**	**-**	**Update**	**P1**	**Write B (B->B1)**	**-**
3	**T2**	**2**	**Update**	**P2**	**Write C (C->C1)**	**-**
4	**T2**	**3**	**Abort**			
5	**Begin checkpoint**					
6	**TT:[(T1,1,active)(T2,4,abort)],DPT:[(P1 2),(P2 3)] End checkpoint**					
7	**T2**	**4**	**CLR**		**Undo T2 LSN 3**	**2**
8	**T2**	**7**	**CLR**		**Undo T2 LSN 2**	**-**
9	**T2**	**8**	**End**			
10	**T3**	**-**	**Update**	**P2**	**Write D (D->D1)**	**-**
11	**T1**	**1**	**Commit**			
12	**T1**	**11**	**End**			
13	**T3**	**10**	**CLR**		**Undo T3 LSN 10**	
14	**T3**	**13**	**End**			

These pages are finally flushed to the disk together with the LOG

PageId	Value	pageLSN
P1	A1 B	8
P2	C D	13