① Normal operation $W_1[A] \rightarrow W_2[B] \rightarrow W_2[C] \rightarrow$ Flush P2 to disk $\rightarrow W_1[D] \rightarrow C_1 \rightarrow W_2[B] \rightarrow$ T1 done committing $\rightarrow$ CRASH

Flush P2 to disk → need to force-write tail of log to disk before page

C_1 → need to force-write tail of log to disk in order to commit T1

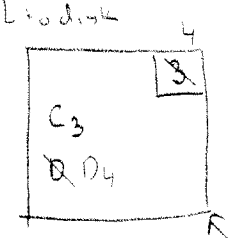
In memory

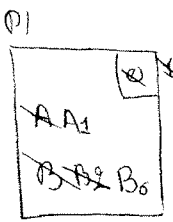
Transaction Table

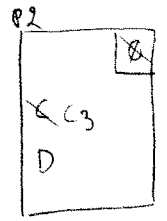
transID	lastLSN	Status
~~T1~~	~~4 5~~	~~R~~ Commited
T2	~~2 3~~ 6	Running

Dirty Page Table

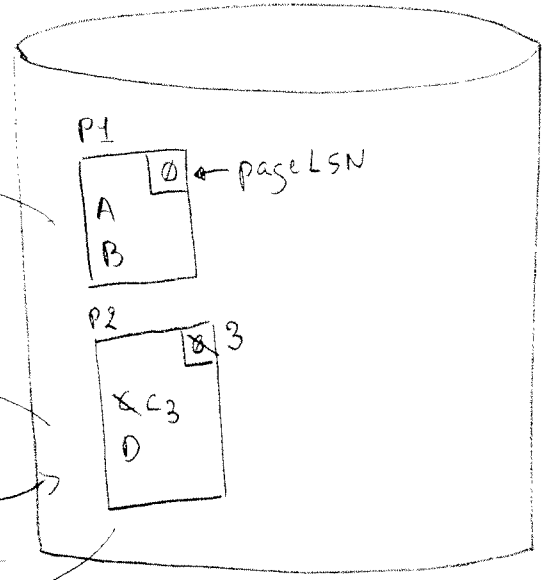
pageID	recLSN
P1	1
~~P2~~	~~3~~ {Flushed to disk
P2	4







First



Second

Log (tail in memory, rest on disk)

Lost after crash

LSN	1	2	3 (write to disk)	4	5 (write to disk)	6	7
transID	T1	T2	T2	T1	T1	T2	T1
prevLSN	—	—	2	1	4	3	5
type	Update	U	U	U	Commit	U	End
logentry	write A $A \rightarrow A_1$	write B $B \rightarrow B_2$	write C $C \rightarrow C_3$	write D $D \rightarrow D_4$	—	write B $B_2 \rightarrow B_6$	—
undoNextLSN (only for CLR)	—	—	—	—	—	—	—

② Analysis phase

During analysis, we encounter transaction records

Transaction Table

trans ID	last LSN	Status
T1	~~1~~ ~~4~~ 5	Unknown ~~Committed~~
T2	~~2~~ 3	U

Dirty Page Table

page ID	rec LSN
P1	1
P2	3

Analysis Rules

- END Record removes transaction From table
- Other records update last LSN
- Commit record changes status to C
- For redoable records update Dirty Page Table

③ Redo phase Repeating history

Start at firstLSN smallest LSN in Dirty Page Table

For each redoable log record (update or CLR), redo if necessary

LSN1: ① check if P1 is in dirty page table YES

② check if recLSN for P1 ≤ LSN1 YES

③ load from disk & check if pageLSN < LSN1 Yes

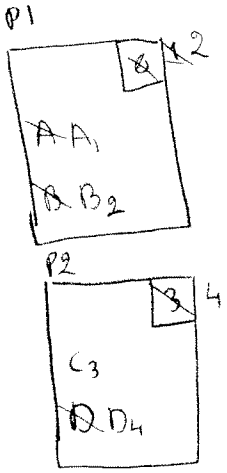
Redo change

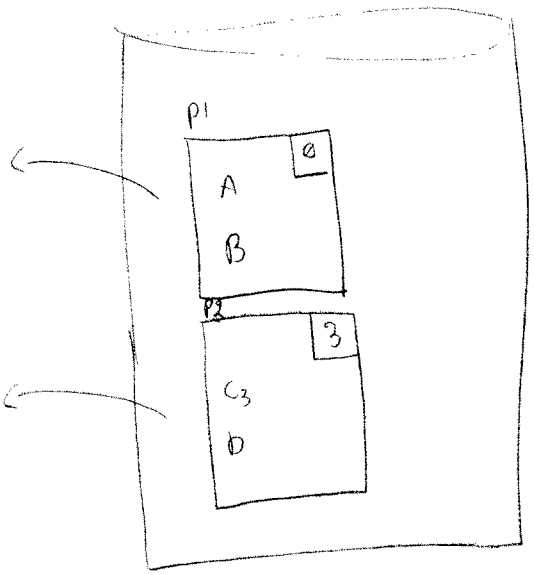
LSN2. Redo change

LSN3: No need to redo

LSN4 Redo

LSN5: Skip





System back in state as of time when log last flushed to disk

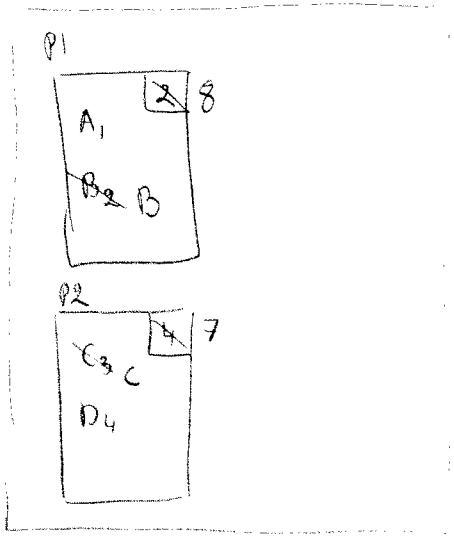
Now write an END type record for T1.

Notice that LSN=6 for this record.

And remove from transaction table

Undo phase Need to Undo T2
Notice no need to write abort log record

SN	6	7	8	9
ansID	T1	T2	T2	T2
·cv LSN	5	—	—	8
ype	E	CLR	CLR	E
'gentry	—	Undo T2 LSN 3	Undo T2 LSN 2	—
dNextLSN	—	2	—	—



Start undo at LSN=3
FOR LSN=3, Write CLR
(Notice that we did not write abort log record)
FOR LSN=2, Write CLR
Write an End log record.

Second crash

① What if we crash again now? IF we did not write anything to disks, we redo analysis/undo/redo exactly in the same way.

② IF we flushed log to disk except for T2's end log record then →

Analysis	T1	5	5	Removed thanks to end record
	T2	8	4	Same dirty page table

Redo same as before until LSN 5
LSN 6 do nothing
LSN 7 if we previously flushed page to disk then do nothing
else apply
LSN 8 same

Undo Finds only CLR without undoNextLSN
So write only an END record

③ IF we had flushed end log record then transactions table would have been empty and we would not undo anything

④ IF we crashed before second CLR
Analysis & Redo same as ②
Add second CLR + END record

2) Checkpoints: 3 steps

Step 1: begin_checkpoint record is written

Step 2: end_checkpoint record is constructed with current content of transaction table and dirty page table. The record is appended to the log.

Step 3: LSN of begin_checkpoint is written in special place

} called a Fuzzy checkpoint

In our example, imagine that we checkpointed after flushing P2 to disk and before $W_1[D]$.

Then after analysis, we would have:

Dirty Page Table	
P1	1
P2	4

Redo would still start at LSN 1

but we would know that LSN 3 did not need to be redone without loading P2 from disk.

And if we had also flushed P1 to disk before the checkpoint we would get the following table after analysis.

Dirty Page Table	
P2	4

Redo would start at LSN = 4

(c) Finally imagine that instead of aborting, T2 aborts ... W2[10] → T2 done committing → Abort T2

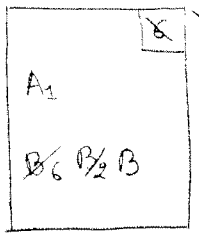
Transaction Table

transID	lastLSN	Status
~~T1~~ removed	~~5~~	~~Committed~~
~~T2~~	~~8~~ 9 10 11	~~Running~~ Aborted

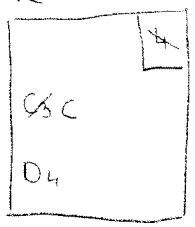
Dirty Page Table

pageID	recLSN
P1	1
P2	4

P1



P2



Log

LSN	6	7	8	9	10	11	12
transID	T2	T1	T2	T2	T2	T2	T2
prevLSN	3	5	6	—	—	—	11
type	U	End	Abort	CLR	CLR	CLR	End
logentry	write B $B_2 \rightarrow B_6$	—	—	Undo T2 LSN 6	Undo T2 LSN 3	Undo T2 LSN 2	—
undoNextLSN	—	—	—	3	2	—	—