

17.4.2

<start U>
<U, A, 10, 11>
<start T>
<T, B, 20, 21>
<U, C, 30, 31>
<T, D, 40, 41>
<commit T>
<U, E, 50, 51>
<commit U>

a) crash etter at <start T> kom på loggdiskene:

Verken T eller U er fullført. Begge skal ruller tilbake.

Disk etter recovery:

A	10
B	
C	
D	
E	

} endres ikke

Logdisk etter recovery:

<start U>
<U, A, 10, 11>
<start T>
<abort U>
<abort T>

b) crash etter at <commit T> kom på loggdiskene:

Løper gjennom loggen bakfra og ruller tilbake ikkecommittede (U),
leper dretter gjennom loggen forover og skriver committede på nytt (T):

Disk etter recovery:

A	10
B	21
C	30
D	41
E	

} endres ikke

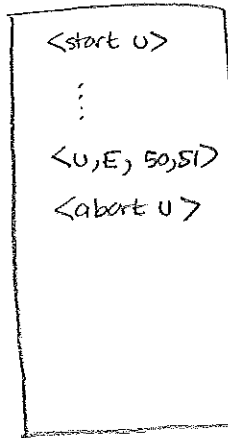
Logdisk etter recovery:

<start U>
<U, A, 10, 11>
<start T>
<T, B, 20, 21>
<U, C, 30, 31>
<T, D, 40, 41>
<commit T>
<abort U>

17.4.2

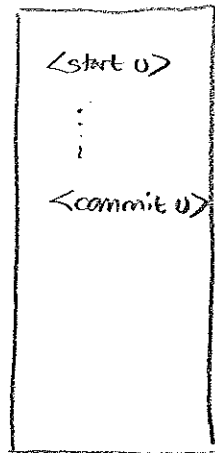
c) Crash etter at $\langle U, E, 50, 51 \rangle$ kom på loggdiskene:

A	10
B	21
C	30
D	41
E	50



d) Crash etter at $\langle \text{commit } U \rangle$ på loggdiskene:

A	11
B	21
C	31
D	41
E	51



18.4.1

b)

(i) sl hvis kun leseaksjon på et element, XL eller, u til slutt?

$sl_1(A); r_1(A); sl_2(B); r_2(B); sl_3(C); r_3(C);$
 $sl_1(B); r_1(B); sl_2(C); r_2(C); sl_3(D); r_3(D);$
 $xl_1(C); w_1(C); u_1(A); u_1(B); u_1(C); xl_2(D); w_2(D)$
 $u_2(B); u_2(C); u_2(D); xl_3(E); u_3(C); u_3(D); u_3(E);$
 $w_3(E)$

(ii)

T_1	T_2	T_3
$sl_1(A)$ $r_1(A)$		
	$sl_2(B)$ $r_2(B)$	
		$sl_3(C)$ $r_3(C)$
$sl_1(B)$ $r_1(B)$		
	$sl_2(C)$ $r_2(C)$	
		$sl_3(D)$ $r_3(D)$
$xl_1(C)$ - vent		
	$xl_2(D)$ - vent	
		$xl_3(E)$ $u_3(C)$ $w_3(E)$ $u_3(D)$ $u_3(E)$
	$xl_2(D)$ $w_2(D)$ $u_2(B)$ $u_2(C)$ $u_2(D)$	
$xl_1(C)$ $w_1(C)$ $u_1(A)$ $u_1(B)$ $u_1(C)$		

(iii)-(vi): Ikke aktuelle;
intet behov for oppgradering
av låser.

18.4.1

c)

(i) $x_1(A); r_1(A); x_2(B); r_2(B); x_3(C); r_3(C);$
 $s_1(B); r_1(B); s_2(C); r_2(C); s_3(A); r_3(A);$
 $w_1(A); u_1(A); u_1(B); w_2(B); u_2(B); u_2(C);$
 $w_3(C); u_3(C); u_3(A);$

(ii)

T_1	T_2	T_3
$x_1(A)$ $r_1(A)$		
	$x_2(B)$ $r_2(B)$	
		$x_3(C)$ $r_3(C)$
$s_1(B)$ -vent		
	$s_2(C)$ -vent	
		$s_3(A)$ -deadlock

(iii) $s_1(A); r_1(A); s_2(B); r_2(B); s_3(C); r_3(C);$
 $s_1(B); r_1(B); s_2(C); r_2(C); s_3(A); r_3(A);$
 $x_1(A); w_1(A); u_1(A); u_1(B); x_2(B); w_2(B);$
 $u_2(B); u_2(C); x_3(C); w_3(C); u_3(C); u_3(B)$

18.4.1

c) (fäts.)

(iv)	<u>T₁</u>	<u>T₂</u>	<u>T₃</u>
	s ₁ (A)		
	r ₁ (A)		
		s ₂ (B)	
		r ₂ (B)	
			s ₃ (C)
			r ₃ (C)
	s ₁ (B)		
	r ₁ (B)		
		s ₂ (C)	
		r ₂ (C)	
			s ₃ (A)
			r ₃ (A)
	x ₁ (A)		
	-vent		
		x ₂ (B)	
		-vent	
			x ₃ (C)
			-vranglös

18.4.1

c) (forts.)

(v) $ul_1(A); r_1(A); ul_2(B); r_2(B); ul_3(C); r_3(C);$
 $sl_1(B); r_1(B); sl_2(C); r_2(C); sl_3(A); r_3(A);$
 $xl_1(A); w_1(A); u_1(A); u_1(B); xl_2(B); w_2(B);$
 $u_2(B); u_2(C); xl_3(C); w_3(C); u_3(C); u_3(A);$

(vi)

$T_1 \quad T_2 \quad T_3$

$ul_1(A)$

$r_1(A)$

$ul_2(B)$

$r_2(B)$

$ul_3(C)$

$r_3(C)$

Anta
prioritet
til skrivere;
da får ingen
fa sl etter
at ul er
fakt

→ $sl_1(B)$
-vent

$sl_2(C)$
-vent

$sl_3(A)$
-vranglås

Alternativt
prioritet
til lesere

$T_1 \quad T_2 \quad T_3$

→ $sl_1(B)$

$r_1(B)$

$sl_2(C)$

$r_2(C)$

$sl_3(A)$

$r_3(A)$

$xl_1(A)$
-vent

$xl_2(B)$
-vent

$xl_3(C)$
-vranglås

e)

- (i) $x_{L1}(A); r_1(A); x_{L2}(B); r_2(B); x_{L3}(C); r_3(C);$
 $s_{L1}(B); r_1(B); s_{L2}(C); r_2(C); s_{L3}(D); r_3(D);$
 $w_1(A); u_1(A); u_1(B); w_2(B); u_2(B); u_2(C);$
 $w_3(C); u_3(C); u_3(D);$

G(i)	T ₁	T ₂	T ₃
	$x_{L1}(A)$ $r_1(A)$		
		$x_{L2}(B)$ $r_2(B)$	
			$x_{L3}(C)$ $r_3(C)$
	$s_{L1}(B)$ - vent		
		$s_{L2}(C)$ - vent	
			$s_{L3}(D)$ $r_3(D)$ $w_3(C)$ $u_3(C)$ $u_3(D)$
		$s_{L2}(C)$ $r_2(C)$ $u_2(B)$ $u_2(C)$	
	$s_{L1}(B)$ $r_1(B)$ $w_1(A)$ $u_1(A)$ $u_1(B)$		

e). (facts.)

(iii) $sl_1(A); r_1(A); sl_2(B); r_2(B); sl_3(C); r_3(C);$
 $sl_1(B); r_1(B); sl_2(C); r_2(C); sl_3(D); r_3(D);$
 $xl_1(A); w_1(A); u_1(A); u_1(B); xl_2(B); w_2(B);$
 $u_2(B); u_2(C); xl_3(C); u_3(C); u_3(D)$

(iv)

T_1	T_2	T_3
$sl_1(A)$		
$r_1(A)$		
	$sl_2(B)$	
	$r_2(B)$	
		$sl_3(C)$
		$r_3(C)$
$sl_1(B)$		
$r_1(B)$		
	$sl_2(C)$	
	$r_2(C)$	
		$sl_3(D)$
		$r_3(D)$
$xl_1(A)$		
$w_1(A)$		
$u_1(A)$		
$u_1(B)$		
	$xl_2(B)$	
	$w_2(B)$	
	$u_2(B)$	
	$u_2(C)$	
		$xl_3(C)$
		$u_3(C)$
		$u_3(D)$

18,4.1

e) (forts.)

(v) $ul_1(A); r_1(A); ul_2(B); r_2(B); ul_3(C); r_3(C);$
 $sl_1(B); r_1(B); sl_2(C); r_2(C); sl_3(D); r_3(D);$
 $xL_1(A); w_1(A); u_1(A); u_1(B); xL_2(B); w_2(B);$
 $u_2(B); u_2(C); xL_3(C); u_3(C); u_3(D)$

(vi)

T_1	T_2	T_3
-------	-------	-------

$ul_1(A)$
 $r_1(A)$

$ul_2(B)$
 $r_2(B)$

$ul_3(C)$
 $r_3(C)$

Antar prioritet til skrivere; da får ingen ta st etter at ut er tatt

$sl_1(B)$
-vent

$sl_2(C)$
-vent

$sl_3(D)$
 $r_3(D)$
 $xL_3(C)$
 $u_3(C)$
 $u_3(D)$

$sl_2(C)$
 $r_2(C)$
 $xL_2(B)$
 $w_2(B)$
 $u_2(B)$
 $u_2(C)$

$sl_1(B)$
 $r_1(B)$
 $xL_1(A)$
 $w_1(A)$
 $u_1(A)$
 $u_1(B)$

Alternativ 1
Prioritet til lesere

T_1	T_2	T_3 (fått)
-------	-------	--------------

$sl_1(B)$
 $r_1(B)$

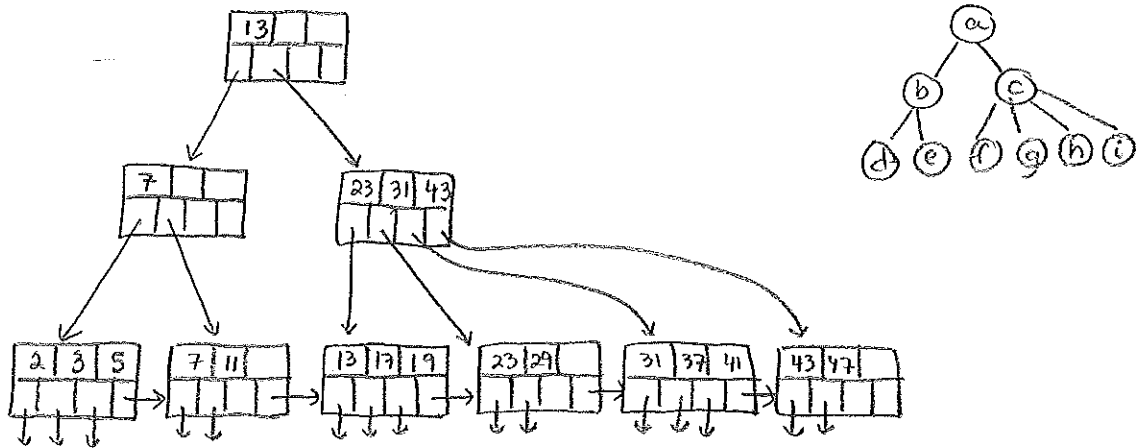
$sl_2(C)$
 $r_2(C)$

$sl_3(D)$
 $r_3(D)$

$xL_1(A)$
 $w_1(A)$
 $u_1(A)$
 $u_1(B)$

$xL_2(B)$
 $w_2(B)$
 $u_2(B)$
 $u_2(C)$

$xL_3(C)$
 $u_3(C)$
 $u_3(D)$



Treprotokollen:

1. I B-trær seltes første lås på roten
2. Videre låser kan tildeles noder hvis foreldrenoden er låst av transaksjonen
3. Låser kan frigis når som helst
4. Låser som er frigitt, kan ikke relåses av samme transaksjon

a) Innsetting av 4: Låser a, b. Ser i b at uansett hva som skjer under b, er det plass i b til en ny peker. Så låser opp a. Låser d. Splitter i to noder og flytter 2,3 på første og 4,5 på andre. Låser opp d. Justerer innholdet i b, låser opp b.

b) Innsetting av 30: Låser a, c, g. Her kan ikke a låses opp ennå fordi vi ikke vet før vi kommer videre ned om innsetningen vil kreve en splitting av et løv og påfølgende splitting av c. Når node g nås og vi ser at det er plass, kan a og c låses opp før g oppdateres. Så oppdateres g og g låses opp.

c) Slett 37: Låser a, c. Ser i c at uansett hva som skjer under c, vil ikke slettingen medføre at c skal slås sammen med en av sine søsken. Så låser opp a. Låser h. Ser at slettingen ikke krever påfølgende sammenslåing av noder, så låser opp c. Endrer i h, låser opp h.

d) Slett 7: Låser a, b. Her kan vi ikke friggi låsen i a ennå, for b inneholder minimalt antall pekere og kan derfor bli slått sammen med c, hullket påvirker også innholdet i a. Låser e. Ser at e kommer under kritisk grense og låser derfor d for å se om en peker i d kan flyttes til e. Det kan den siden d er over nedre grense. Straks vi vet dette, vet vi også at det ikke blir noen sammenslåinger av noder, så låser på a kan frigis. b må imidlertid også justeres, så vi holder låsen i denne. Innholdet i d justeres, låsen til d frigis. Innholdet i b justeres, låsen til b frigis. Innholdet i e justeres, låsen frigis.

Telest
forholder seg
til det
oppriinnelige
treet, ikke
slite det ser
ut eller plot. a)

18.8.1

a)

T_1	T_2	$RT(A)$	$WT(A)$	$C(A)$	$RT(B)$	$WT(B)$	$C(B)$
		t_0	u_0	true	v_0	w_0	true
st_1							
$r_1(A)$		st_1					
	st_2						
	$w_2(B)$						
	$r_2(A)$	st_2				st_2	false

skriver for sent $\rightarrow w_1(B)$

- Vent til $C(B)$ er sann, sjekk situasjonen på nytt

- Vekkes opp, bruk Thomas' skriverregel og gjør ingen endringer.

c_2 Eller c_2 gjøres umiddelbart etter $r_2(A)$. Da vil fortsatt $w_1(B)$ komme for sent. Anvend Thomas' skriverregel: La T_1 fortsette uten å skrive B.

b)

T_1	T_2	$RT(A)$	$WT(A)$	$C(A)$	$RT(B)$	$WT(B)$	$C(B)$
		t_0	u_0	true	v_0	w_0	true
st_1							
	st_2						
$r_1(A)$		st_1					
	$r_2(B)$						
	$w_2(A)$		st_2	false		st_2	

$w_1(B)$
- null tilbake; T_2 har alt lest B.