

HW8 Solution

1

a)

No, U will not be able to obtain lock for x and will wait until T finishes.

Transaction T	Transaction U
openTransaction a = read (x) Lock x write (z,2) Lock z d = read (z) closeTransaction Release lock x, z	openTransaction b = read (y) Lock y c = read (x) Can not obtain lock for x write (z,1) write (x,5) closeTransaction

Transaction T	Transaction U
openTransaction <div style="border: 1px solid blue; border-radius: 50%; padding: 5px; display: inline-block;"> a = read (x) write (z,2) </div> Remove d = read (z) closeTransaction	openTransaction b = read (y) c = read (x) write (z,1) write (x,5) closeTransaction

b)

No, U cannot obtain writelock for z

Transaction T	Transaction U
openTransaction a = read (x) <u>readlock x</u> e = read (z) <u>readlock z</u> write(x,5) <u>writelock x</u> write (y,2) closeTransaction	openTransaction c = read (y) <u>readlock y</u> write (z,1) can not obtain <u>writelock z</u> write(y,3) closeTransaction

Transaction T	Transaction U
openTransaction a = read (x) e = read (z) write(x,5) write (y,2) closeTransaction	openTransaction c = read (y) <u>write (z,1)</u> Change to read(z) write(y,3) closeTransaction

2

a) Yes

b) Yes

c) Yes

d) Yes

3

Forward:

V validates first: V's write set: {w}, U's read set: {w}, T's read set: {v}. Overlap, V aborts.

U validates: U's write set: {w}, T's read set: {v}. No overlap, U commits.

T validates: No other active transactions, T commits.

Backward:

V validates first: No one committed during my transaction, V commits.

U validates: U's read set: {w}, V's write set: {w}. Overlap, U aborts.

T validates: T's read set: {v, w}, ~~U's write set: {w}~~, V's write set: {w}. Overlap, T aborts.

U has aborted. T should not consider U's write set.