

CS 430 Concurrency worksheet

- 1) Apply the appropriate locks and show the resulting schedule for the following sequence of operations using strict 2PL. Assume locks can be upgraded.

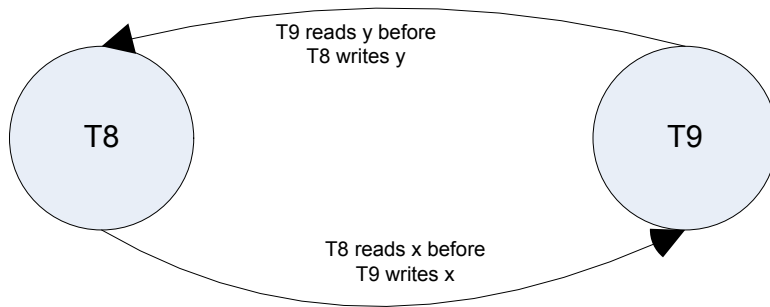
T1:R(X); T2:W(Y); T3:R(X); T2:R(X); T2:R(Z); T2:Commit; T3:W(X); T3:Commit; T1:W(Y); Commit

| T1     | T2     | T3              |
|--------|--------|-----------------|
| RL(x)  |        |                 |
| R(x)   |        |                 |
|        | WL(y)  |                 |
|        | W(Y)   |                 |
|        |        | RL(x)           |
|        |        | R(x)            |
|        | RL(x)  |                 |
|        | R(x)   |                 |
|        | RL(z)  |                 |
|        | R(z)   |                 |
|        | Commit |                 |
|        | U(z)   |                 |
|        | U(x)   |                 |
|        | U(y)   |                 |
|        |        | WL(x) – holds   |
| WL(y)  |        |                 |
| W(y)   |        |                 |
| Commit |        |                 |
| U(y)   |        |                 |
| U(x)   |        |                 |
|        |        | WL(x) – granted |
|        |        | W(x)            |
|        |        | Commit          |
|        |        | U(x)            |

2) Graph the following schedule. Is it conflict serializable?

T8:R(X); T9:R(Y); T8:W(Y); T9:W(X);T8:Commit; T9:Commit;

No, this schedule is not conflict serializable.



- 3) Using the timestamp method of creating conflict serialization (and not using the Thomas write rule), fill in the following chart. Which (if any) transactions are rolled back. T1's timestamp is 3, T2's timestamp is 4.

| Schedule       | Action | R-TS (X) | W-TS(X) | R-TS(Y) | W-TS(Y) |
|----------------|--------|----------|---------|---------|---------|
| Initial Values |        | 1        | 2       | 2       | 1       |
| T1             | R(Y)   |          |         |         |         |
| T2             | R(X)   |          |         |         |         |
| T2             | R(Y)   |          |         |         |         |
| T1             | R(X)   |          |         |         |         |
| T2             | W(Y)   |          |         |         |         |
| T1             | W(X)   |          |         |         |         |
| T2             | W(X)   |          |         |         |         |
| T1             | R(X)   |          |         |         |         |

| Schedule       | Action | R-TS (X)        | W-TS(X)     | R-TS(Y) | W-TS(Y) |
|----------------|--------|-----------------|-------------|---------|---------|
| Initial Values |        | 1               | 2           | 2       | 1       |
| T1             | R(Y)   |                 |             | 3       |         |
| T2             | R(X)   | 4               |             |         |         |
| T2             | R(Y)   |                 |             | 4       |         |
| T1             | R(X)   | 4               |             |         |         |
| T2             | W(Y)   |                 |             |         | 4       |
| T1             | W(X)   | TS(t1) < RTS(X) | Rolled back |         |         |
| T2             | W(X)   |                 | 4           |         |         |
| T1             | R(X)   |                 |             |         |         |

- 4) Now let's do another timestamp problem, only this time with the Thomas write rule. T1's timestamp is 3, T2's timestamp is 4.

| Schedule       | Action | R-TS (X) | W-TS(X) | R-TS(Y) | W-TS(Y) |
|----------------|--------|----------|---------|---------|---------|
| Initial Values |        | 1        | 2       | 2       | 1       |
| T1             | R(Y)   |          |         |         |         |
| T2             | R(X)   |          |         |         |         |
| T1             | R(X)   |          |         |         |         |
| T2             | W(Y)   |          |         |         |         |
| T1             | W(Y)   |          |         |         |         |
| T2             | W(X)   |          |         |         |         |
| T1             | R(X)   |          |         |         |         |

| Schedule       | Action | R-TS (X)       | W-TS(X) | R-TS(Y) | W-TS(Y) |
|----------------|--------|----------------|---------|---------|---------|
| Initial Values |        | 1              | 2       | 2       | 1       |
| T1             | R(Y)   |                |         | 3       |         |
| T2             | R(X)   | 4              |         |         |         |
| T1             | R(X)   | TS not changed |         |         |         |
| T2             | W(Y)   |                |         |         | 4       |
| T1             | W(Y)   |                |         |         | Ignored |
| T2             | W(X)   |                | 4       |         |         |
| T1             | R(X)   |                |         |         |         |

5) What if T2 had read Y first?

| Schedule       | Action | R-TS (X) | W-TS(X) | R-TS(Y) | W-TS(Y) |
|----------------|--------|----------|---------|---------|---------|
| Initial Values |        | 1        | 2       | 2       | 1       |
| T1             | R(Y)   |          |         |         |         |
| T2             | R(Y)   |          |         |         |         |
| T1             | R(X)   |          |         |         |         |
| T2             | W(Y)   |          |         |         |         |
| T1             | W(Y)   |          |         |         |         |
| T2             | W(X)   |          |         |         |         |
| T1             | R(X)   |          |         |         |         |

| Schedule       | Action | R-TS (X) | W-TS(X) | R-TS(Y) | W-TS(Y) |
|----------------|--------|----------|---------|---------|---------|
| Initial Values |        | 1        | 2       | 2       | 1       |
| T1             | R(Y)   |          |         | 3       |         |
| T2             | R(Y)   |          |         | 4       |         |
| T1             | R(X)   | 3        |         |         |         |
| T2             | W(Y)   |          |         |         | 4       |
| T1             | W(Y)   |          |         |         | Abort   |
| T2             | W(X)   |          | 4       |         |         |
| T1             | R(X)   |          |         |         |         |

6) Now let's take a look at multiversion TS protocol

T1:W(x) (10)      T1 writes a value of 10 into x  
 T3:R(x)            T3 reads the value of 10 from x  
 T3:W(x) (8)        T3 writes a value of 8 into x  
 T2:R(x)            T2 reads a value of 10 from x  
 T4:R(x)            T4 reads a value of 8 from x  
 T4:W(x) (4)        T4 writes a value 4 to x  
 T2:W(x)            T2 aborts because it is attempting to write a value that T3 should have read

T1:W(x) (10)

| Cache Item | RTS | WTS | Value |
|------------|-----|-----|-------|
| x          |     | 1   | 10    |

T3:R(x)

| Cache Item | RTS | WTS | Value |
|------------|-----|-----|-------|
| x          | 3   | 1   | 10    |

T3:W(x) (8)

| Cache Item | RTS | WTS | Value |
|------------|-----|-----|-------|
| x          | 3   | 1   | 10    |
| x          |     | 3   | 8     |

T2:R(x)

| Cache Item | RTS | WTS | Value |
|------------|-----|-----|-------|
| x          | 3   | 1   | 10    |
| x          |     | 3   | 8     |

T4:R(x)

| Cache Item | RTS | WTS | Value |
|------------|-----|-----|-------|
| x          | 3   | 1   | 10    |
| x          | 4   | 3   | 8     |

T4:W(x) (4)

| Cache Item | RTS | WTS | Value |
|------------|-----|-----|-------|
| x          | 3   | 1   | 10    |
| x          | 4   | 3   | 8     |
| x          |     | 4   | 4     |