

Transactions - Bachelor Project - MFF UK 2007

File Protocol Without Locks 2PL Protocol Creating Schedule Simulation Edit Tree Diagram Tree Lock Protocol Create Schedule-Tree LP

UP UP UP SIMULATE TILL THE END Wait-Die Protocol Lock Protocol
 DOWN DOWN DOWN SIMULATE STEP BY STEP

| T0 | T1 | T2 | T0 | T0 | T0 | T0 |
|---------------|-------------------------------|--|---------------|---------------|---------------|---|
| X(B) ABORT | X(A) WRITE(A) UNLOCK(A) | S(B) READ(B) X(B) WRITE(B) UNLOCK(B) | X(B) ABORT | X(B) ABORT | X(B) ABORT | X(B) WRITE(B) S(A) READ(A) UNLOCK(B) UNLOCK(A) |

Ready

Final Schedule

| T0 | T1 | T2 | T0* | T0** | T0*** | T0**** |
|---------------|-------------------------------|-----------------|---------------|------|-------|--------|
| X(B) ABORT | X(A) WRITE(A) UNLOCK(A) | S(B) READ(B) | X(B) ABORT | | | |

Start Total Commander 6.5... WindowsApplication1... Prezentace - Microsof... Transactions - Bachel... Final Schedule 18:05

Transactions - Bachelor Project - MFF UK 2007

File Protocol Without Locks 2PL Protocol Create Schedule Simulation Edit Tree Diagram Tree Lock Protocol Create Schedule-Tree LP Deadlock Detection Help

UP UP UP SIMULATE TILL THE END
 DOWN DOWN DOWN SIMULATE STEP BY STEP

| T0 | T1 | T2 | T1 | T0 | T1 | T0 | T1 |
|----------------------------------|----------------------------------|---|---------------|---------------|---------------|------|--|
| S(A) READ(A) X(A) ABORT | S(A) READ(A) S(B) ABORT | X(B) S(A) READ(A) X(A) WRITE(A) UNLOCK(B) UNLOCK(A) | S(A) ABORT | S(A) ABORT | S(A) ABORT | S(A) | S(A) READ(A) X(A) WRITE(A) UNLOCK(A) |
| | | | | | | | S(A) READ(A) S(B) READ(B) X(B) WRITE(B) UNLOCK(A) UNLOCK(B) |

Ready

Start Total Commander 6.5... WindowsApplication1... Prezentace - Microsof... Transactions - Bachel... 16:29

Transactions - Bachelor Project - MFF UK 2007

File Protocol Without Locks 2PL Protocol Create Schedule Simulation Edit Tree Diagram Tree Lock Protocol Create Schedule-Tree LP Deadlock Detection Help

UP DOWN UP DOWN UP DOWN SIMULATE TILL THE END SIMULATE STEP BY STEP

| T0 | T1 | T2 | T2 | T2 | T2 |
|-----------------------------------|--|---------------|---------------|---------------|---|
| S(A) R(A) X(A) UNLOCK(A) | S(A) X(B) WRITE(B) UNLOCK(A) UNLOCK(B) | X(B) ABORT | X(B) ABORT | X(B) ABORT | X(B) S(A) READ(A) X(A) WRITE(A) UNLOCK(B) UNLOCK(A) |

Ready

Start Total Commander 6.5... WindowsApplication1 ... Presentace - Microsof... Transactions - Bachel...

Transactions - Bachelor Project - MFF UK 2007

File Protocol Without Locks 2PL Protocol Create Schedule Simulation Edit Tree Diagram Tree Lock Protocol Create Schedule-Tree LP Deadlock Detection Help

UP DOWN UP DOWN UP DOWN SIMULATE TILL THE END SIMULATE STEP BY STEP

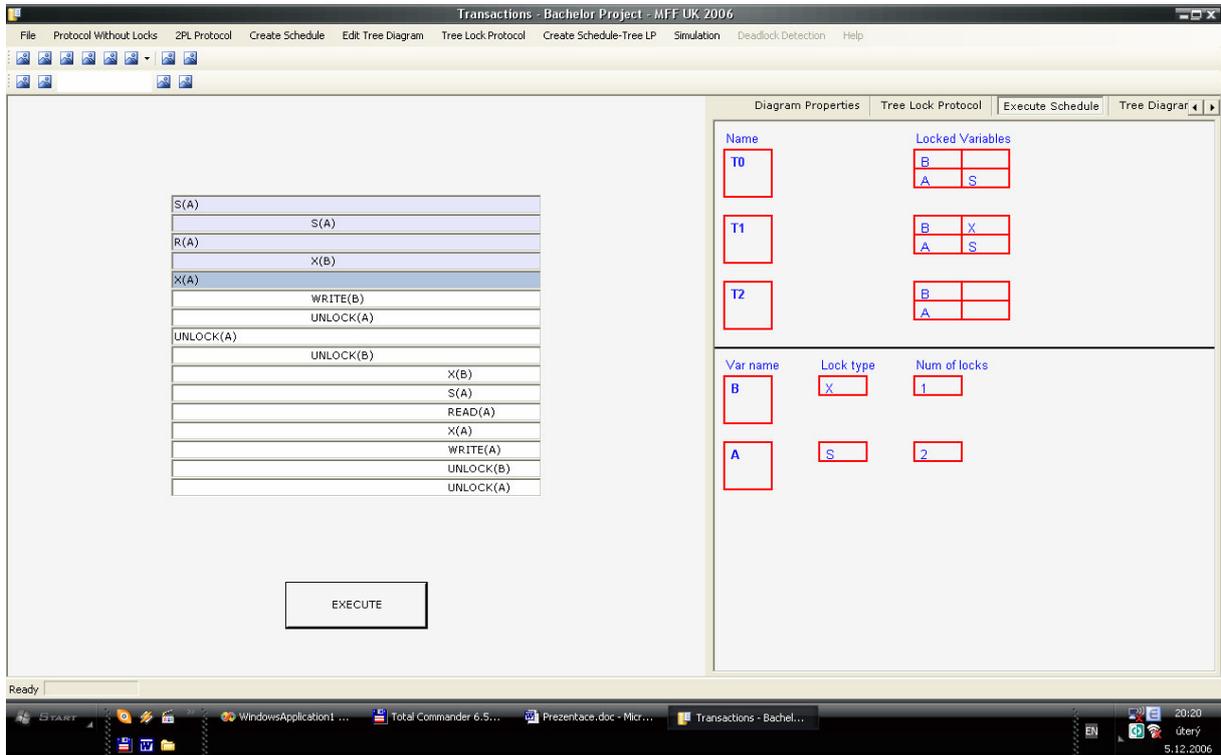
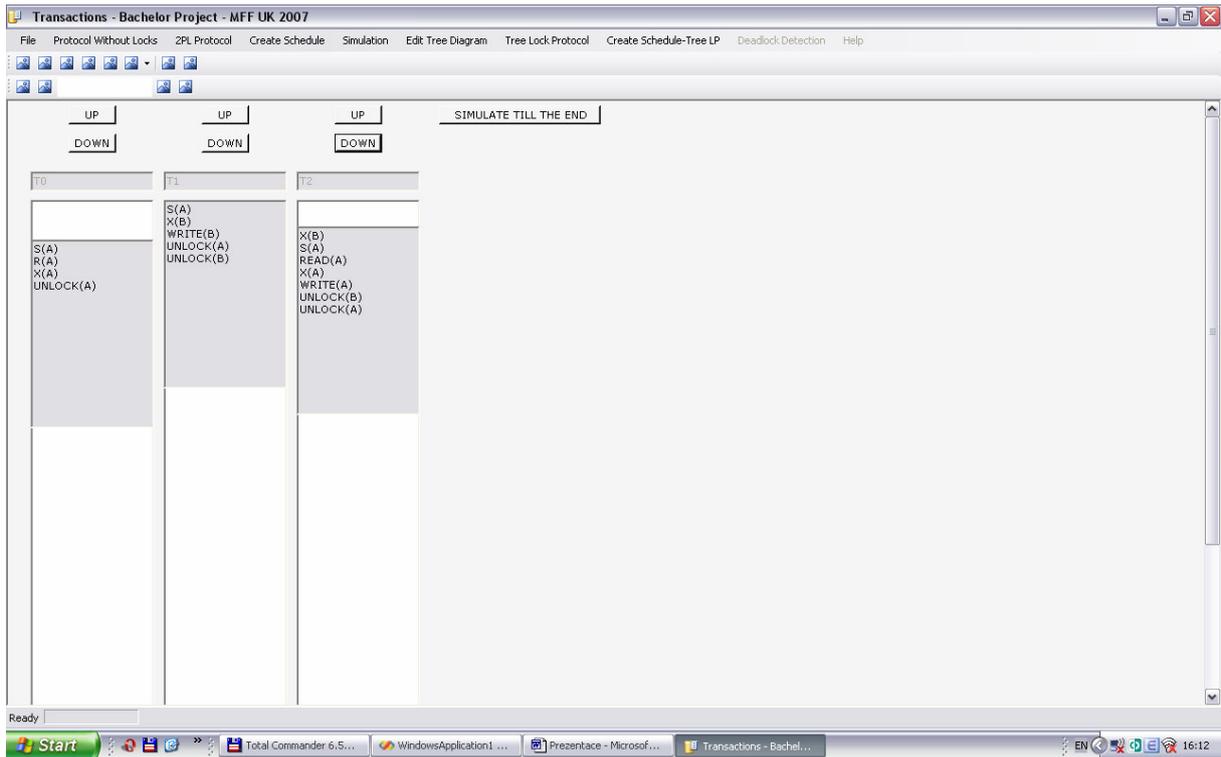
| T0 | T1 | T2 |
|-----------------------------------|--|---|
| S(A) R(A) X(A) UNLOCK(A) | S(A) X(B) WRITE(B) UNLOCK(A) UNLOCK(B) | X(B) S(A) READ(A) X(A) WRITE(A) UNLOCK(B) UNLOCK(A) |

Final Schedule

| T0 | T1 | T2 | T2* | T2** | T2*** |
|-----------|--------------|---------------|---------------|---------------|---|
| S(A) | S(A) X(B) | X(B) ABORT | X(B) ABORT | X(B) ABORT | X(B) S(A) READ(A) X(A) WRITE(A) UNLOCK(B) UNLOCK(A) |
| R(A) | WRITE(B) | | | | |
| X(A) | UNLOCK(A) | | | | |
| UNLOCK(A) | UNLOCK(B) | | | | |

Ready

Start Total Commander 6.5... WindowsApplication1 ... Presentace - Microsof... Transactions - Bachel... Final Schedule



Transactions - Bachelor Project - MFF UK 2006

File Protocol Without Locks 2PL Protocol Create Schedule Edit Tree Diagram Tree Lock Protocol Create Schedule-Tree LP Simulation Deadlock Detection Help

Protocol Without Locks Well-formed Control Two-phased Control Lock Protoc...

| T0 | T1 | T2 |
|-----------------------------------|--|---|
| S(A) R(A) X(A) | S(A) X(B) WRITE(B) | X(B) S(A) READ(A) X(A) WRITE(A) |
| S(A) R(A) X(A) UNLOCK(A) | S(A) X(B) WRITE(B) UNLOCK(A) UNLOCK(B) | X(B) S(A) READ(A) X(A) WRITE(A) UNLOCK(B) UNLOCK(A) |

| T0 | T1 | T2 | T2* | T2** | T2*** |
|------------|------------|--------------------------|---------------|---------------|---|
| S(A) | S(A) | X(B) | | | |
| R(A) | X(B), WAIT | S(A) | | | |
| X(A), WAIT | WRITE(B) | READ(A) X(A) ABORT | | | |
| | UNLOCK(A) | | X(B) ABORT | | |
| UNLOCK(A) | UNLOCK(B) | | | X(B) ABORT | |
| | | | | | X(B) S(A) READ(A) X(A) WRITE(A) UNLOCK(B) UNLOCK(A) |

Ready

WindowsApplication1 ... Total Commander 6.5... Prezentace.doc - Micr... Transactions - Bachel... Final Schedule

20:18 Oterý 5.12.2006

Transactions - Bachelor Project - MFF UK 2006

File Protocol Without Locks 2PL Protocol Create Schedule Edit Tree Diagram Tree Lock Protocol Create Schedule-Tree LP Simulation Deadlock Detection Help

Well-formed Control Two-phased Control Lock Protocol Diagram Properties

| T0 | T1 | T2 | T3 |
|---------------------|---------------------|---------------------|---------------------|
| READ(A) WRITE(A) | READ(B) WRITE(B) | | |
| READ(B) WRITE(B) | | READ(B) WRITE(B) | READ(A) WRITE(A) |

Well-formed Criteria

- Schedule is legal
- Transaction locks an attribute, if it wants to use it
- Transaction DOES NOT lock an attribute, if already done by itself
- Transaction DOES NOT unlock an attribute, if it was NOT locked by itself
- At the end of a transaction, attributes locked by itself are unlocked

Transactions are NOT well formed

Ready

WindowsApplication1 ... Total Commander 6.5... Prezentace.doc - Micr... Transactions - Bachel...

20:15 Oterý 5.12.2006

Transactions - Bachelor Project - MFF UK 2006

File Protocol Without Locks 2Phased Lock Protocol Create 2Phased Schedule Edit Tree Diagram Tree Lock Protocol Create Schedule-Tree LP Simulate Help

| T0 | T1 | T2 | T3 |
|---|---|---------------------------------|---------------------------------|
| READ(B) READ(D) READ(E) | READ(A) READ(B) READ(F) | READ(A) | READ(B) |
| LOCK(B) READ(B) LOCK(D) READ(D) LOCK(E) READ(E) UNLOCK(B) UNLOCK(D) UNLOCK(E) | LOCK(A) READ(A) LOCK(B) READ(B) LOCK(F) READ(F) UNLOCK(A) UNLOCK(B) UNLOCK(F) | LOCK(A) READ(A) UNLOCK(A) | LOCK(B) READ(B) UNLOCK(B) |
| LOCK(B) READ(B) LOCK(D) READ(D) LOCK(E) READ(E) UNLOCK(B) | LOCK(A) READ(A) LOCK(B) READ(B) | | |

Ready

```

graph TD
    s["s [2]"] --> b["b [3]"]
    s --> c["c"]
    b --> d["d [2]"]
    b --> e["e [1]"]
    b --> f["f"]
    d --> g["g"]
    d --> h["h [1]"]
    e --> i["i"]
    h --> j["j"]
  
```

Lock Protocol Diagram Properties Tree Lock Protocol Tree Diagram

<Shift>: move the diagram
<Alt>: add a new child node

Windows taskbar: 10:59

Transactions - Bachelor Project - MFF UK 2006

File Protocol Without Locks 2Phased Lock Protocol Create 2Phased Schedule Edit Tree Diagram Tree Lock Protocol Create Schedule-Tree LP Simulate Help

| T0 | T1 | T2 | T3 | T4 |
|---|--------------------|---|---|---------------------------------|
| LOCK(A) READ(A) | LOCK(B) READ(B) | LOCK(C) READ(C) UNLOCK(C) | | |
| LOCK(C) READ(C) WRITE(A) WRITE(C) UNLOCK(A) | LOCK(A) READ(A) | WRITE(B) LOCK(C) READ(C) WRITE(C) UNLOCK(A) | LOCK(A) READ(A) WRITE(A) UNLOCK(A) | |
| UNLOCK(C) | UNLOCK(B) | | | LOCK(B) READ(B) UNLOCK(B) |
| | UNLOCK(C) | | | |

Ready

Well-formed Control Two-phased Control Lock Protocol Diagram Properties

Lock Protocol Criteriaes

- Schedule well-formed
- Schedule two-phased

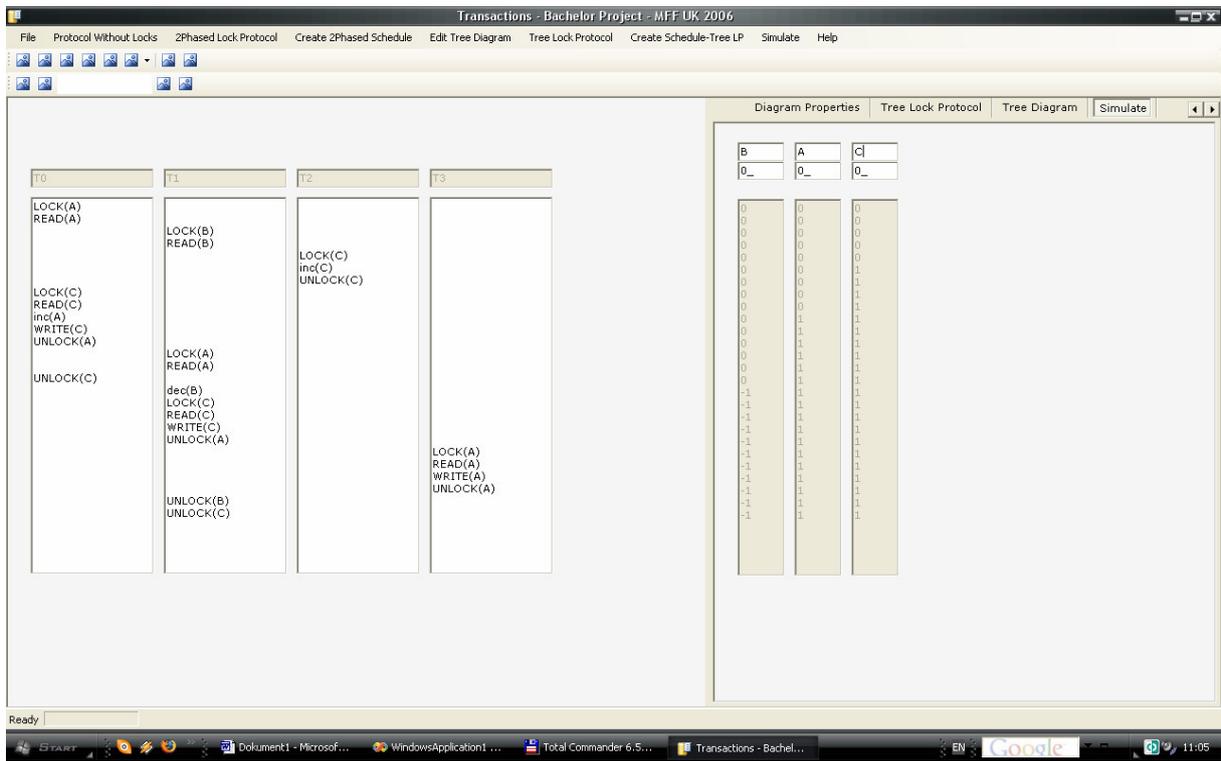
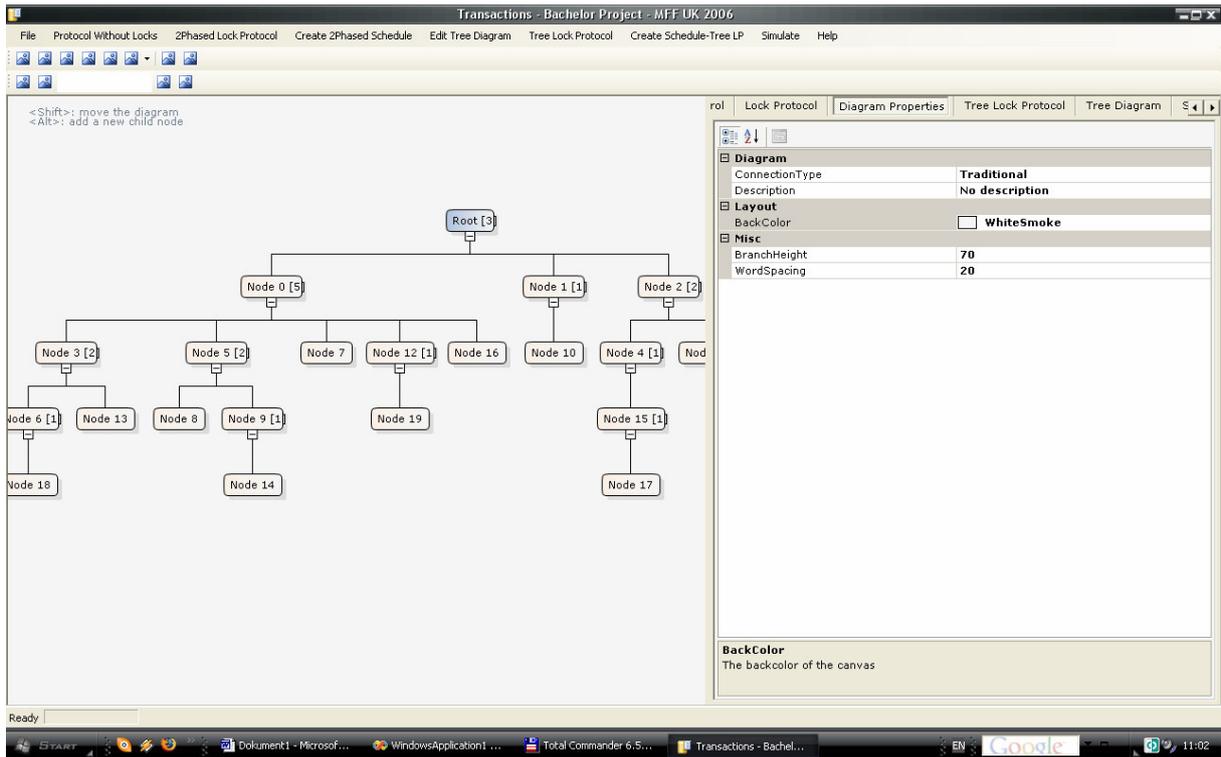
Schedule is serializable: {T2,T0,T1,T4,T3}

Schedule is serializable: {T2,T0,T1,T4,T3}

```

graph TD
    T2 --> T0
    T2 --> T1
    T2 --> T3
    T0 --> T1
    T1 --> T4
    T3 --> T4
  
```

Windows taskbar: 11:01

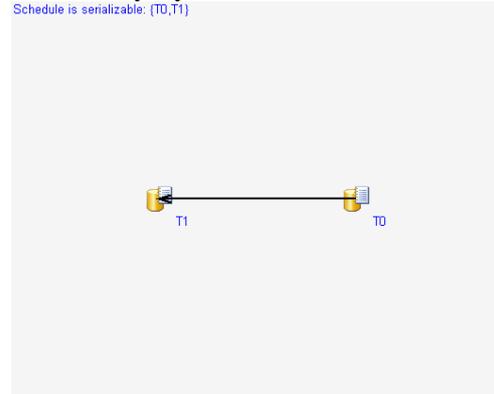


Testovací data

Testy serializovatelnosti
Protocol bez zamku:
schedule_serial.txt

read(a)
 action(a)
 write(a)
 read(a)
 action(a)
 write(a)
 read(b)
 action(b)
 write(b)
 read(b)
 action(b)
 write(b)

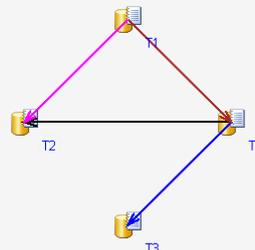
Zobrazeny vysledek



schedule serial1.txt

read(a)
 write(a)
 read(b)
 write(b)
 read(b)
 write(b)
 read(b)
 write(b)
 read(a)
 write(a)

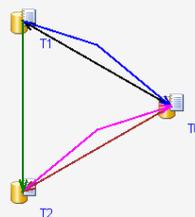
Schedule is serializable: {T1,T0,T3,T2}



schedule not serial.txt

read(a)
 read(a)
 write(a)
 read(a)
 write(a)
 write(a)

Schedule is NOT serializable!



Protocol se zamky:

schedule2.txt

testovani funkcnosti importu pro ruzne paznaky

```
loc K( A)
read(a)
      lock(b)
      read(b)

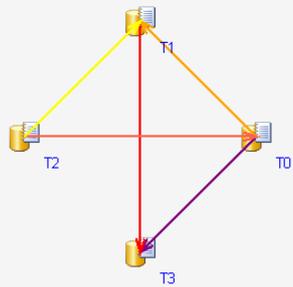
      lock(c)
      read(c)
      unlock(c)

lock(c)
read(c)
write(a)
write(c)
unlock(a)
      lock(A)
      read(a)
unlock(c)
      write(b)
      lock(c)
      read(c)
      write(c)
      unlock(a)

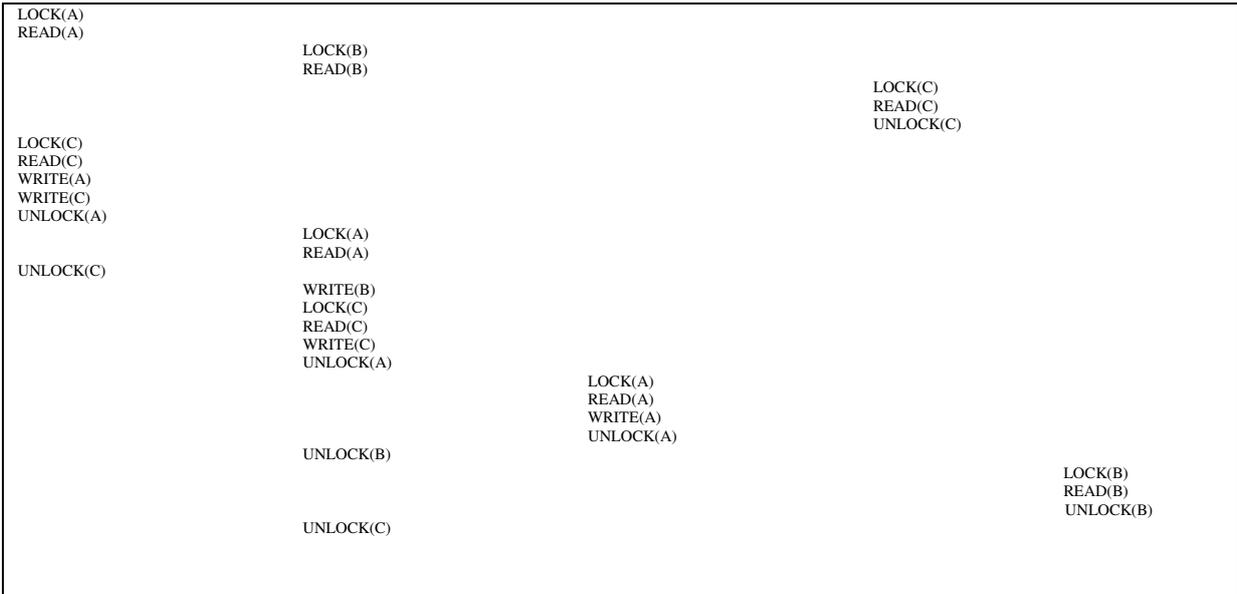
      lock(a)
      read(a)
      write(a)
      unlock(A)

unlock(b)
unlock(c)
```

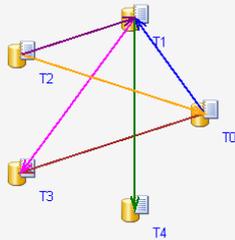
Schedule is serializable: {T2,T0,T1,T3}



schedule3.txt

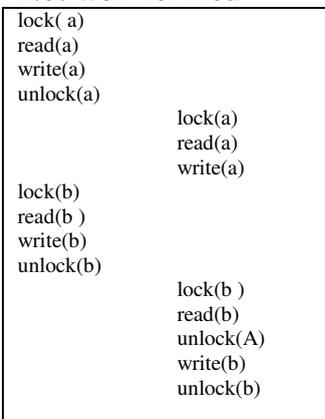


Schedule is serializable: {T2,T0,T1,T4,T3}



schedule with locks.txt

Not well-formed



**Vytvareni rozvrhu:
Bez prevence uvaznuti**

transactions.txt

READ(A)
READ(C)
WRITE(A)
WRITE(C)

READ(B)
READ(A)
WRITE(B)
READ(C)
WRITE(C)

READ(A)
WRITE(A)

READ(C)

--

vysledek:

LOCK(A)
READ(A)

LOCK(B)
READ(B)

LOCK(C)
READ(C)
UNLOCK(C)

LOCK(C)
READ(C)
WRITE(A)
WRITE(C)
UNLOCK(A)

LOCK(A)
READ(A)

UNLOCK(C)

WRITE(B)
LOCK(C)
READ(C)
WRITE(C)
UNLOCK(A)

LOCK(A)
READ(A)
WRITE(A)
UNLOCK(A)

UNLOCK(B)
UNLOCK(C)

--

s prevenci uvaznuti

transactions 2PL deadlock prevent1.txt

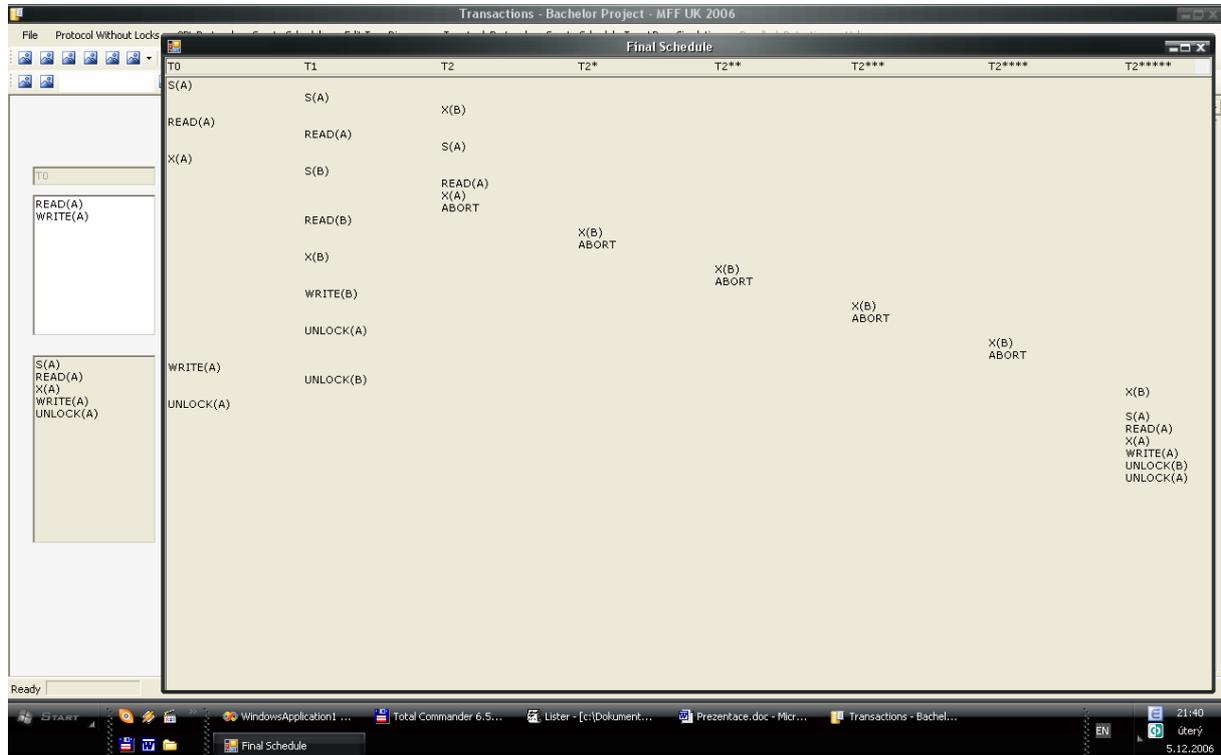
s(A)
r(a)
x(a)

S(A)
X(B)
write(B)

X(B)
S(A)

X(A)
write(A)

vysledek



Stromovy protocol

not serial:

lock(d)
read(d)
unlock(d)
 lock(b)
 read(b)
 unlock(b)
lock(h)
read(h)
 lock(e)
 read(e)
unlock(h)
 unlock(e)
lock(b)
read(b)
unlock(b)

serial:

```
lock(b)
    lock(d)
    lock(h)
    unlock(d)
lock(e)
unlock(e)
lock(d)
unlock(b)
    lock(b)
    lock(e)
    unlock(h)
lock(g)
unlock(d)
    lock(d)
    lock(h)
    unlock(d)
    unlock(h)
    unlock(e)
    unlock(b)
unlock(g)
```

```
dia1
schedule tree protocol not ser1
schedule tree protocol not ser
```

Testovani serializovatelnosti

```
diagram tree serializable1.gxl
schedule tree protocol serializable1.txt
```

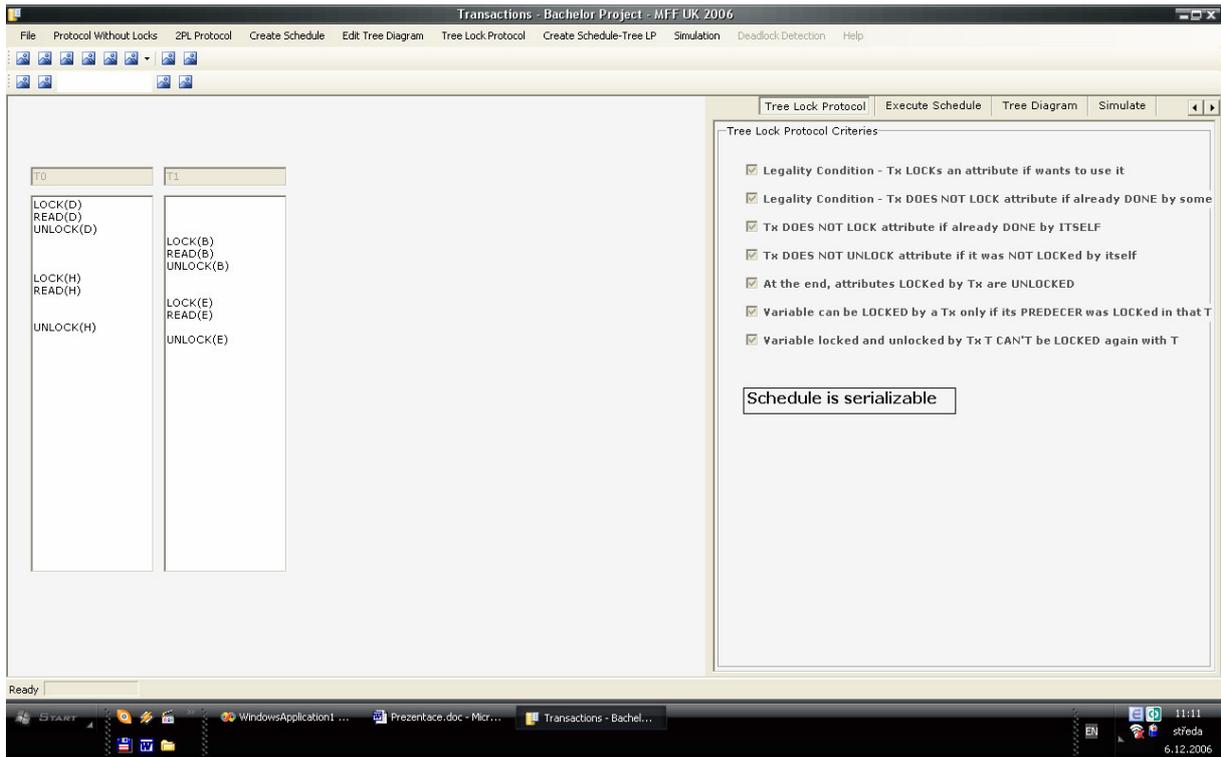
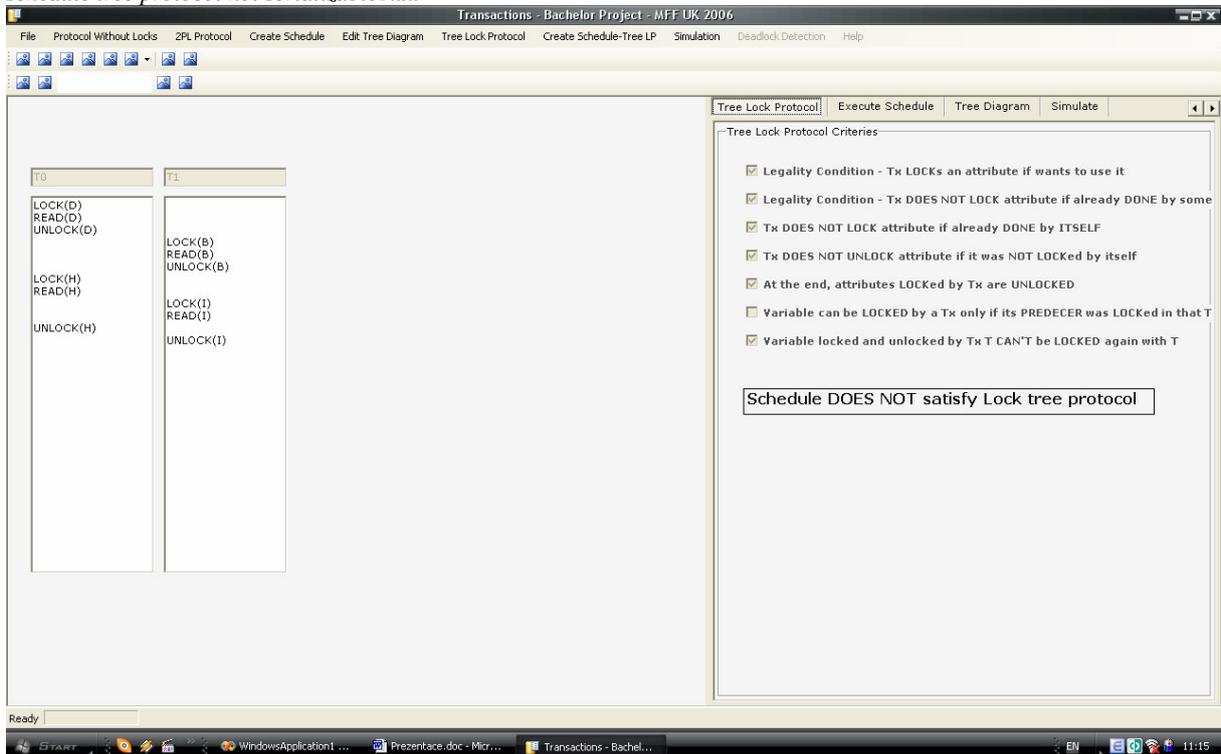
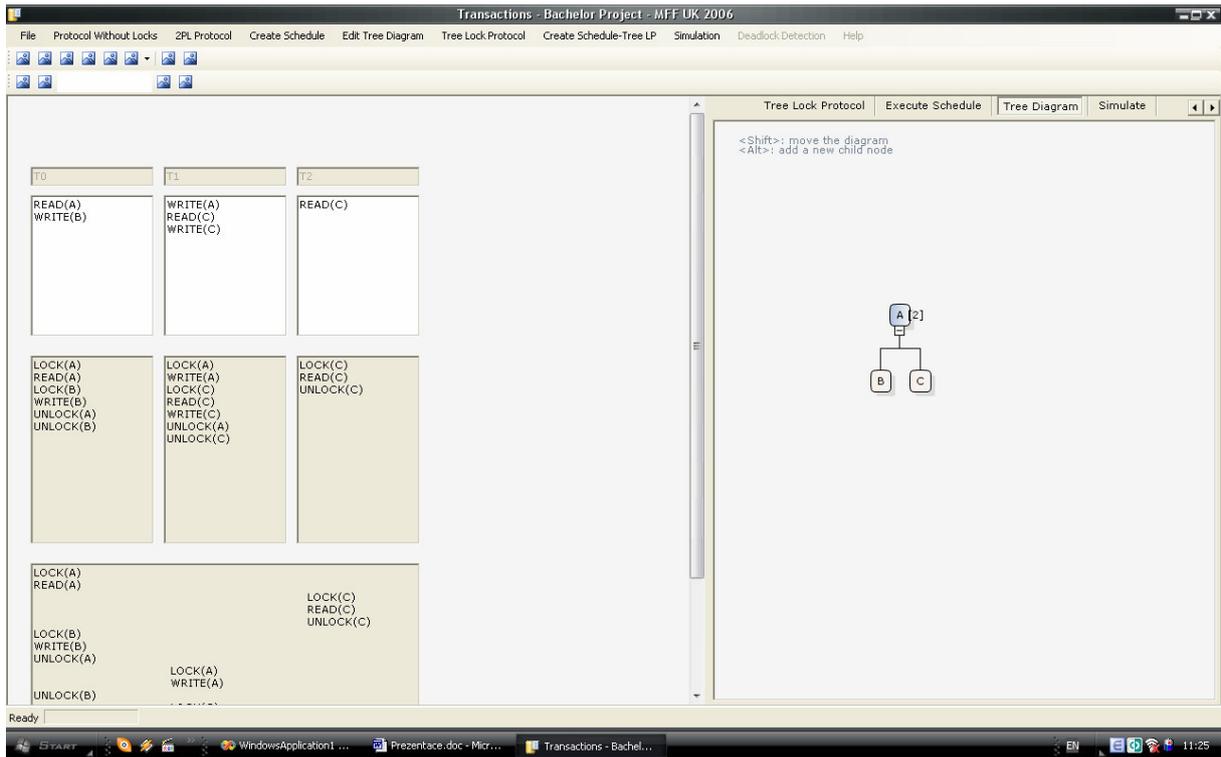


diagram tree serializable1.gxl
schedule tree protocol not serializable1.txt



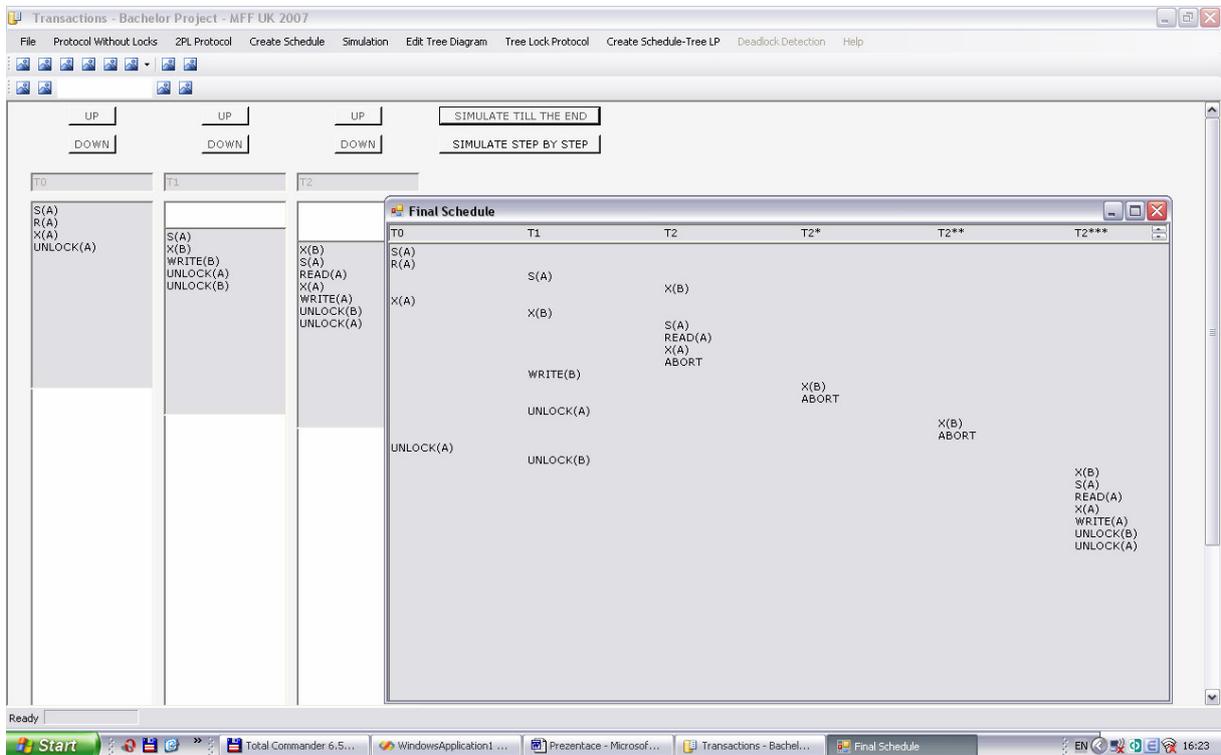
Vytaveni rozvrhu

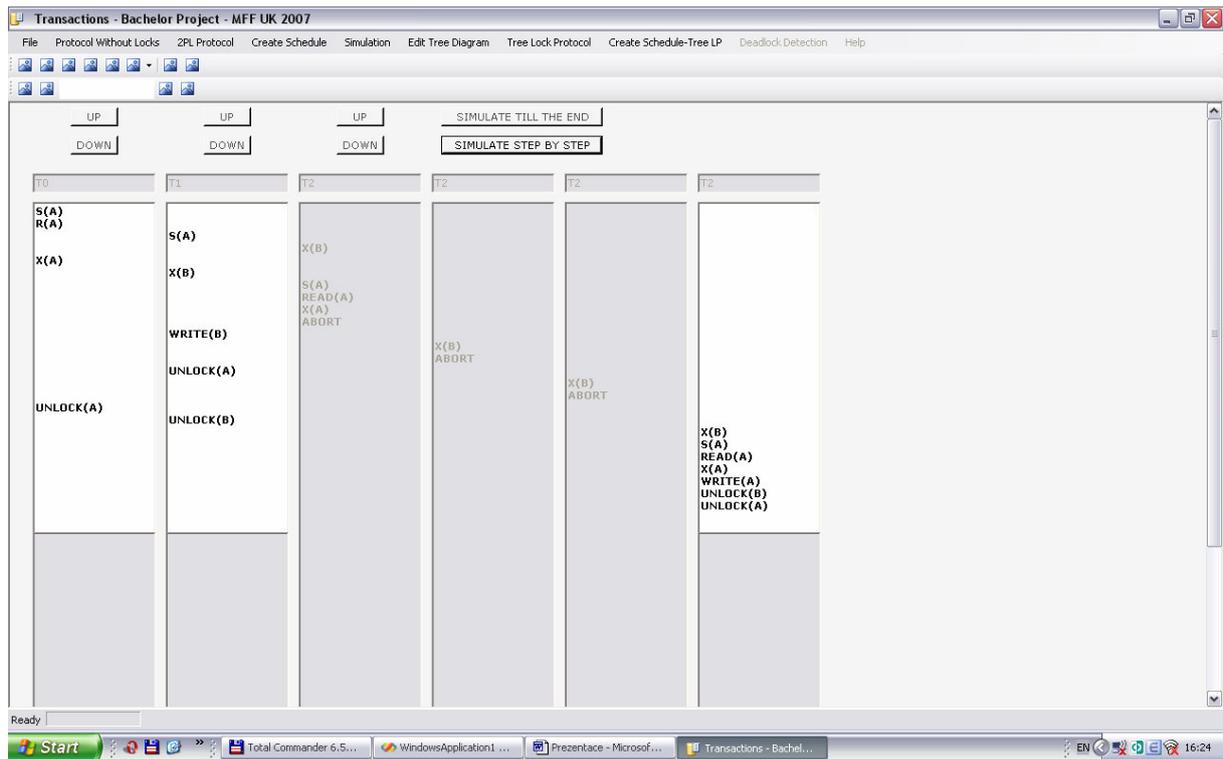
diagram1.gxl
transactions TP ok1.txt



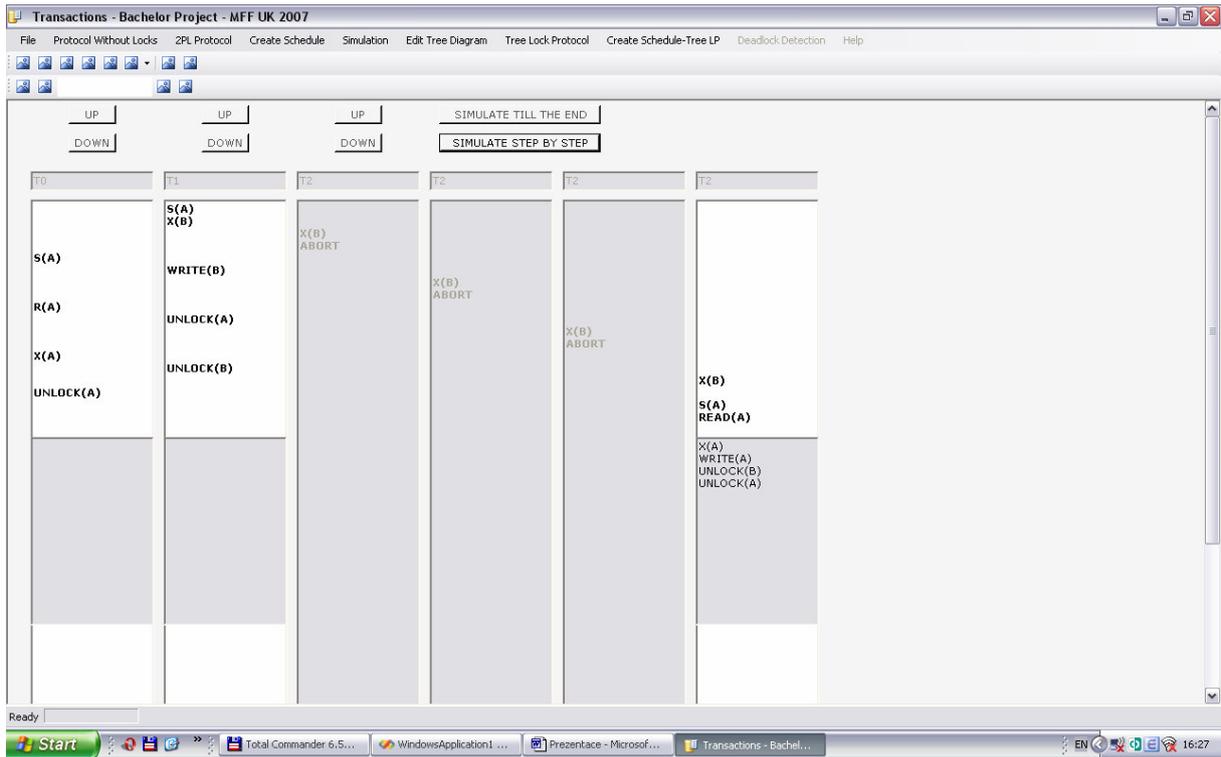
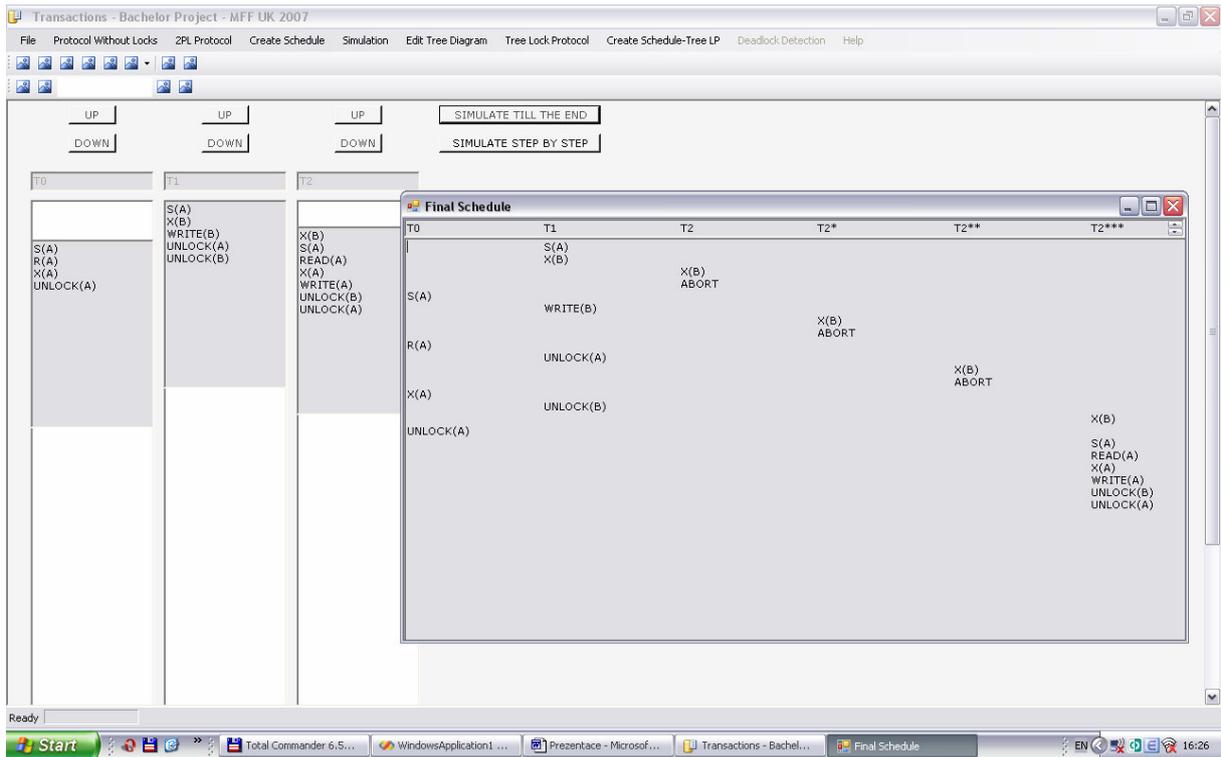
Simulace vytvoreni rozvrhu (s moznosti zadani zacatku posunutim transakci):

transactions 2PL deadlock prevent1:
 T0(0), T1(2), T2(3)





transactions 2PL deadlock prevent1:
T0(3), T1(0), T2(2)



Lock Protocol s moznym posunem pocatecniho timestampu
transactions lock protocol.txt

Zachyceni deadlock:
transactions lock protocol deadlock.txt