CSE 5331 **Summer 2019**

Project 1- Phase 2

Team Details: Harsh Sharma (1001642088)

Shitiz Kumar Aggarwal (1001669578)

Programming Language Used: Python

Steps to be followed to run the code are:

- Install latest version of python in your system (python 3.7).
- Install pycharm community edition which will run the source code.
- Create a folder in pycharmprojects named twoPl.
- Paste single input file while executing given by professor in the folder and run the code against the particular order like input1, input2, input3 and son till input7 one by one.
- You will get the respective output against each input given by the professor.

DESCRIPTION:

Steps followed to perform Rigorous 2 Phase Locking using wound-wait protocol dealing with deadlock:

The program will read the schedule from a text file line by line and is going to recognize the trans_ id, operations (read/write operations) and the data items.

- When **'b'** is encountered it will call the begin transaction function and will make the entry in the transaction table in Transaction ID field.
- When **'r'** is encountered it will check all the records in the Lock Table. If the item is not in the lock table its unlocked and read lock is enforced to it and the record is updated in the transaction table and also entry is made into lock table. If the item is present in lock table, then check its state in the table, if the state is 'read' then append the Trans_ID in lock table if it is 'write' then check its timestamp and apply wound-wait concept. If the state is blocked check item in lock table, if it is present then add the Trans_ID in the waiting list. If not present then make the entry in the lock table and change its state to active in transaction table and update the lock table. If the state is found to be aborted or committed then do nothing.
- When **'w'** is encountered it will make changes in the lock table and will add items in the transaction table and the table will be updated. While writing any transaction we need to check the status of the transaction. If the state is active then check the data item in lock table, if present and state is 'read' and only one transaction present then upgrade the lock to write else check the timestamp and apply wound-wait concept. If state is 'write' then check the timestamp and apply wound-wait concept and if the data item is not present then the entry will be made in the table. If the state is aborted or committed then do nothing.
- When **'e'** is encountered the changes will be made and status of the transaction will be checked. If the state is active then all the locks are released from the transaction table and state will be changed to commit in transaction table. Then the next transaction in the queue will be

processed. If the state is blocked we will check the lock state if the locks are released we will apply the locks later and will change the state to commit in transaction table.

Outputs:

Input 1:

```
b1;
r1 (Y);
w1 (Y);
r1 (Z);
b2;
r2 (X);
w2 (X);
w1 (Z);
e1;
r2 (Y);
b3;
r3 (Z);
w3 (Z);
w2 (Y);
e2;
b4;
w4 (Y);
r3 (X);
w3 (X);
e3;
w4 (X);
e4;
```

Output1:

```
Beginning of the Transaction 1
Read Operation
Read lock is enforced on data item Y by Transaction 1
Write Operation
Read lock is upgraded to Write Lock on data item Y for transaction 1
Read Operation
Read lock is enforced on data item Z by Transaction 1
Beginning of the Transaction 2
Read Operation
Read lock is enforced on data item X by Transaction 2
Write Operation
Read lock is upgraded to Write Lock on data item X for transaction 2
Write Operation
Read lock is upgraded to Write Lock on data item Z for transaction 1
End
```

Transaction Committed 1
Unlock all the resources held by transaction 1
checking for transactions if they are waiting for free resources
Read Operation
Read lock is enforced on data item Y by Transaction 2
Beginning of the Transaction 3
Read Operation
Read lock is enforced on data item Z by Transaction 3
Write Operation
Read lock is upgraded to Write Lock on data item Z for transaction 3
Write Operation
Read lock is upgraded to Write Lock on data item Y for transaction 2
End
Transaction Committed 2
Unlock all the resources held by transaction 2
checking for transactions if they are waiting for free resources
Beginning of the Transaction 4
Write Operation
Write Lock enforced on data item Y by Transaction 3
Read Operation
Read lock is enforced on data item X by Transaction 3
Write Operation
Read lock is upgraded to Write Lock on data item X for transaction 3
End
Transaction Committed 3
Unlock all the resources held by transaction 3
checking for transactions if they are waiting for free resources
Write Operation
Write Lock enforced on data item X by Transaction 4
End
Transaction Committed 4
Unlock all the resources held by transaction 4
checking for transactions if they are waiting for free resources

Process finished with exit code 0

Input 2:

b1;
r1 (Y);
w1 (Y);
r1 (Z);
b2;
r2 (X);
w2 (X);
w1 (Z);
r2 (Y);

e1;
b3;
r3 (Z);
w3 (Z);
w2 (Y);
e2;
r3 (X);
w3 (X);
b4;
w4(X);
e3;
e4;

Output 2:

Beginning of the Transaction 1
Read Operation
Read lock is enforced on data item Y by Transaction 1
Write Operation
Read lock is upgraded to Write Lock on data item Y for transaction 1
Read Operation
Read lock is enforced on data item Z by Transaction 1
Beginning of the Transaction 2
Read Operation
Read lock is enforced on data item X by Transaction 2
Write Operation
Read lock is upgraded to Write Lock on data item X for transaction 2
Write Operation
Read lock is upgraded to Write Lock on data item Z for transaction 1
Read Operation
Write lock conflict as : data item Y has Read Lock by Transaction 1
Wound-Wait is called
Transaction state for transaction 2 changed to block
End
Transaction Committed 1
Unlock all the resources held by transaction 1
checking for transactions if they are waiting for free resources
Attempting Operation r2 (Y);

Read Operation
Read lock is enforced on data item Y by Transaction 2
Beginning of the Transaction 3
Read Operation
Read lock is enforced on data item Z by Transaction 3
Write Operation
Read lock is upgraded to Write Lock on data item Z for transaction 3
Write Operation
Read lock is upgraded to Write Lock on data item Y for transaction 2

End
Transaction Committed 2
Unlock all the resources held by transaction 2
checking for transactions if they are waiting for free resources
Read Operation
Read lock is enforced on data item X by Transaction 3
Write Operation
Read lock is upgraded to Write Lock on data item X for transaction 3
Beginning of the Transaction 4
Write Operation
Write Lock Conflict: data item X is held by Write Lock for Transaction 3
wound wait called
Transaction state for transaction 4 changed to block
End
Transaction Committed 3
Unlock all the resources held by transaction 3
checking for transactions if they are waiting for free resources
Attempting Operation w4(X);

Write Operation
End
Transaction Committed 4
Unlock all the resources held by transaction 4
checking for transactions if they are waiting for free resources

Process finished with exit code 0

Input 3:

b1;
r1 (Y);
w1 (Y);
r1 (Z);
b2;
r2 (X);
w2 (X);
w1 (Z);
r2 (Y);
e1;
b3;
r3 (Z);
w3 (Z);
w2 (Y);
e2;

b4;
w4(X);
r3 (X);
e4;
w3 (X);
e3;

Output 3:

Beginning of the Transaction 1
Read Operation
Read lock is enforced on data item Y by Transaction 1
Write Operation
Read lock is upgraded to Write Lock on data item Y for transaction 1
Read Operation
Read lock is enforced on data item Z by Transaction 1
Beginning of the Transaction 2
Read Operation
Read lock is enforced on data item X by Transaction 2
Write Operation
Read lock is upgraded to Write Lock on data item X for transaction 2
Write Operation
Read lock is upgraded to Write Lock on data item Z for transaction 1
Read Operation
Write lock conflict as : data item Y has Read Lock by Transaction 1
Wound-Wait is called
Transaction state for transaction 2 changed to block
End
Transaction Committed 1
Unlock all the resources held by transaction 1
checking for transactions if they are waiting for free resources
Attempting Operation r2 (Y);
Read Operation
Read lock is enforced on data item Y by Transaction 2
Beginning of the Transaction 3
Read Operation
Read lock is enforced on data item Z by Transaction 3
Write Operation
Read lock is upgraded to Write Lock on data item Z for transaction 3
Write Operation
Read lock is upgraded to Write Lock on data item Y for transaction 2
End

Transaction Committed 2
Unlock all the resources held by transaction 2
checking for transactions if they are waiting for free resources
Beginning of the Transaction 4
Write Operation
Write Lock enforced on data item X by Transaction 3
Read Operation
Write lock conflict as : data item X has Read Lock by Transaction 4
Wound-Wait is called
Transaction 4 aborted
Unlock all the resources held by transaction 4
checking for transactions if they are waiting for free resources
Attempting Operation r3 (X);
Read Operation
Read lock is enforced on data item X by Transaction 3
End
Transaction 4 aborted
Write Operation
Write Lock enforced on data item X by Transaction 3
End
Transaction Committed 3
Unlock all the resources held by transaction 3
checking for transactions if they are waiting for free resources
Process finished with exit code 0

Input 4:

b1;
r1 (Z);
b2;
r2 (X);
w2 (X);
w1 (Z);
r2 (Y);
r1 (Y);
w1 (Y);
b3;
r3 (Z);
e1;
w3 (Z);
w2 (Y);
e2;
r3 (X);
w3 (X);
e3;

Output 4:

Beginning of the Transaction 1
Read Operation
Read lock is enforced on data item Z by Transaction 1
Beginning of the Transaction 2
Read Operation
Read lock is enforced on data item X by Transaction 2
Write Operation
Read lock is upgraded to Write Lock on data item X for transaction 2
Write Operation
Read lock is upgraded to Write Lock on data item Z for transaction 1
Read Operation
Read lock is enforced on data item Y by Transaction 2
Read Operation
Read lock is enforced on data item Y by Transaction 2
Write Operation
Transaction state for transaction 1 changed to block
Beginning of the Transaction 3
Read Operation
Write lock conflict as : data item Z has Read Lock by Transaction 1
Wound-Wait is called
Transaction state for transaction 3 changed to block
End
Transaction Committed 1
Unlock all the resources held by transaction 1
checking for transactions if they are waiting for free resources
Attempting Operation w1 (Y);

Write Operation
data item Y is under Read Lock by more than one transaction.
Wound Wait is called
Transaction 2 aborted
Unlock all the resources held by transaction 2
checking for transactions if they are waiting for free resources
Attempting Operation w1 (Y);

Write Operation
Attempting Operation w1 (Y);

Write Operation
Write Lock Conflict: data item Y is held by Write Lock for Transaction 1
wound wait called
Transaction state for transaction 1 changed to block
Attempting Operation r3 (Z);

Read Operation

Read lock is enforced on data item Z by Transaction 3
Attempting Operation w1 (Y);

Write Operation
Write Lock Conflict: data item Y is held by Write Lock for Transaction 1
wound wait called
Transaction state for transaction 1 changed to block
Attempting Operation w1 (Y);

Write Operation
Write Lock Conflict: data item Y is held by Write Lock for Transaction 1
wound wait called
Transaction state for transaction 1 changed to block
Write Operation
Read lock is upgraded to Write Lock on data item Z for transaction 3
Write Operation
Write Lock Conflict: data item Y is held by Write Lock for Transaction 1
wound wait called
Transaction state for transaction 2 changed to block
End
Transaction 2 aborted
checking for transactions if they are waiting for free resources
Attempting Operation w1 (Y);

Write Operation
Write Lock enforced on data item Y by Transaction 3
Attempting Operation w2 (Y);

Write Operation
Write Lock Conflict: data item Y is held by Write Lock for Transaction 1
wound wait called
Transaction state for transaction 2 changed to block
Read Operation
Read lock is enforced on data item X by Transaction 3
Write Operation
Read lock is upgraded to Write Lock on data item X for transaction 3
End
Transaction Committed 3
Unlock all the resources held by transaction 3
checking for transactions if they are waiting for free resources

Process finished with exit code 0

Input 5:

b1;
r1 (Y);
r1 (Z);

b2;
r2 (Y);
b3;
r3 (Y);
w1 (Z);
w3 (Y);
e1;
e3;
w2 (Y);
r2 (X);
w2 (X);
e2;

Output 5:

Beginning of the Transaction 1
Read Operation
Read lock is enforced on data item Y by Transaction 1
Read Operation
Read lock is enforced on data item Z by Transaction 1
Beginning of the Transaction 2
Read Operation
Read lock is enforced on data item Y by Transaction 1
Beginning of the Transaction 3
Read Operation
Read lock is enforced on data item Y by Transaction 1
Write Operation
Read lock is upgraded to Write Lock on data item Z for transaction 1
Write Operation
Transaction state for transaction 3 changed to block
End
Transaction Committed 1
Unlock all the resources held by transaction 1
checking for transactions if they are waiting for free resources
Attempting Operation w3 (Y);

Write Operation
Transaction state for transaction 3 changed to block
End
Transaction Committed 3
Unlock all the resources held by transaction 3
checking for transactions if they are waiting for free resources
Attempting Operation w3 (Y);

Write Operation
data item Y is under Read Lock by more than one transaction.
Wound Wait is called
Transaction state for transaction 3 changed to block

Write Operation
Read lock is upgraded to Write Lock on data item Y for transaction 2
Read Operation
Read lock is enforced on data item X by Transaction 2
Write Operation
Read lock is upgraded to Write Lock on data item X for transaction 2
End
Transaction Committed 2
Unlock all the resources held by transaction 2
checking for transactions if they are waiting for free resources
Attempting Operation w3 (Y);

Write Operation

Process finished with exit code 0

Input 6:

b1;
r1 (Y);
r1 (Z);
b2;
r2 (Y);
b3;
r3 (Y);
w1 (Z);
w3 (Y);
w2 (Y);
r2 (X);
e1;
e3;
w2 (X);
e2;

Output 6:

Beginning of the Transaction 1
Read Operation
Read lock is enforced on data item Y by Transaction 1
Read Operation
Read lock is enforced on data item Z by Transaction 1
Beginning of the Transaction 2
Read Operation

Read lock is enforced on data item Y by Transaction 1
Beginning of the Transaction 3
Read Operation
Read lock is enforced on data item Y by Transaction 1
Write Operation
Read lock is upgraded to Write Lock on data item Z for transaction 1
Write Operation
Transaction state for transaction 3 changed to block
Write Operation
Transaction state for transaction 2 changed to block
Read Operation
Read lock is enforced on data item X by Transaction 2
End
Transaction Committed 1
Unlock all the resources held by transaction 1
checking for transactions if they are waiting for free resources
Attempting Operation w3 (Y);
Write Operation
Transaction state for transaction 3 changed to block
Attempting Operation w2 (Y);
Write Operation
Transaction 3 aborted
Unlock all the resources held by transaction 3
checking for transactions if they are waiting for free resources
Attempting Operation w2 (Y);
Write Operation
Readl ock is upgraded to Write Lock on data item Y for transaction 2
Attempting Operation w2 (Y);
Write Operation
Write Lock Conflict: data item Y is held by Write Lock for Transaction 2
wound wait called
Transaction state for transaction 2 changed to block
Attempting Operation w2 (Y);
Write Operation
Write Lock Conflict: data item Y is held by Write Lock for Transaction 2
wound wait called
Transaction state for transaction 2 changed to block
End
Transaction 3 aborted
Attempting Operation w2 (Y);
Write Operation
Write Lock enforced on data item Y by Transaction 2
Write Operation
Read lock is upgraded to Write Lock on data item X for transaction 2

End
Transaction Committed 2
Unlock all the resources held by transaction 2
checking for transactions if they are waiting for free resources
Process finished with exit code 0

Input 7:

b1;
r1 (Y);
w1 (Y);
r1 (Z);
b2;
r2 (Y);
b3;
r3 (Z);
w1 (Z);
w2 (Y);
r2 (X);
e1;
w3 (Z);
e3;
w2 (X);
e2;

Output 7:

Beginning of the Transaction 1
Read Operation
Read lock is enforced on data item Y by Transaction 1
Write Operation
Read lock is upgraded to Write Lock on data item Y for transaction 1
Read Operation
Read lock is enforced on data item Z by Transaction 1
Beginning of the Transaction 2
Read Operation
Write lock conflict as : data item Y has Read Lock by Transaction 1
Wound-Wait is called
Transaction state for transaction 2 changed to block
Beginning of the Transaction 3
Read Operation
Read lock is enforced on data item Z by Transaction 1
Write Operation

Transaction 3 aborted
Unlock all the resources held by transaction 3
checking for transactions if they are waiting for free resources
Attempting Operation r2 (Y);
Read Operation
Write lock conflict as : data item Y has Read Lock by Transaction 1
Wound-Wait is called
Transaction state for transaction 2 changed to block
Attempting Operation w1 (Z);
Write Operation
Read lock is upgraded to Write Lock on data item Z for transaction 1
Write Operation
Write Lock Conflict: data item Y is held by Write Lock for Transaction 1
wound wait called
Transaction state for transaction 2 changed to block
Read Operation
Read lock is enforced on data item X by Transaction 2
End
Transaction Committed 1
Unlock all the resources held by transaction 1
checking for transactions if they are waiting for free resources
Attempting Operation r2 (Y);
Read Operation
Read lock is enforced on data item Y by Transaction 2
Attempting Operation w2 (Y);
Write Operation
Read lock is upgraded to Write Lock on data item Y for transaction 2
Write Operation
Write Lock enforced on data item Z by Transaction 2
End
Transaction 3 aborted
Write Operation
Read lock is upgraded to Write Lock on data item X for transaction 2
End
Transaction Committed 2
Unlock all the resources held by transaction 2
checking for transactions if they are waiting for free resources
Process finished with exit code 0

Refrences:

- https://stackoverflow.com/questions/29722886/2pl-rigorous-vs-strict-model-is-there-any-benefit/30049871
- https://www.geeksforgeeks.org/dbms-concurrency-control-protocol-two-phase-locking-2-pl-ii/
- https://github.com/shubhang-arora/two-phase-locking/
- https://www.youtube.com/watch?v=gJa2uVQDesU
- https://github.com/fernsmark/2-Phase-Locking-Protocol