**2208-CSE-5331-001**

**DBMS MODELS AND IMPLEMENTATION**

**The University of Texas at Arlington**

**The Department of Computer Science and Engineering**

**Snigdha Kethireddy Geethika Lingamaneni**

**1001757794 1001743466**

**Overall Status:**

We have completed implementation of the mentioned methods in the zgt\_tm.c and zgt\_tx.c We are able to generate the output logfiles.

**readtx():** Here , we only use the shared lock. We start the operation and set the lock if the set\_lock() method returns true value then we can lock it and the read operation can be performed.

**writetx():** Here , we only use the exclusive lock. We start the operation and set the lock if the set\_lock() method returns true value then we can lock it and the write operation can be performed.

**aborttx():** Here the transaction’s status will be changed to abort. We use the do\_commit\_abort() method which frees the locks and also prints if transaction is aborted if status is ‘A’.

**committx():** Here the transaction’s status will be changed to commit. We use the do\_commit\_abort() method which frees the locks and also prints if transaction is commited if status is ‘C’ .

**do\_commit\_abort():** This method is used by both aborttx() and committx(). When there is a transaction, it frees the locks and checks if any transaction is waiting with zgt\_nwait(). Here we allot commit if the status is ‘C’ and abort if the status is ‘A’

**perform\_readWrite():** It is used by the set\_lock() as it updates the array by decrementing for reading and incrementing for writing

**set\_lock():** Initially we check if the obj is in hashtable or not and if its not present we add it to the hash table. If it is already present and tx->tid==tpointer->tid then we perform read/write. We check for the type of lock requested, if it has shared lock or exclusive lock. If it is in wait state then we set the semaphore to tid and lock it. Now when the status is active we perform read/write on it and release it.

**Difficulties we encountered:**

We have encountered difficulty initially for running the code, which one might be better to run either Cygwin or omega and we then proceeded with omega. We encountered difficulties with the set\_lock method about how to set the semaphores and how to consider the lock conditions etc. We also had to revise the previous c/c++ concepts to understand the given code.

**File Descriptions:**

We had all the files from the assignment. There were no additional files added or provided. We only had to make changes to some methods in zgt\_tm.c and zgt\_tx.c

**Screen shots of one input file given and achieved output for that:**

Input file: ddlk\_3Txs.txt

A picture containing text

Description automatically generated

Output file achieved: ddlk\_3Txs.log

A picture containing text

Description automatically generated

**Division of Labor:**

It took couple of days for us to understand the concepts and go through the skeletal code.

We then discussed how to proceed with the project and spent 2-3 hours a day initially and then we spent almost like 3-5 hours a day in figuring out what to be done and how to be done. We spent most of the time to understand the set\_lock method.

* readtx, writetx, perform\_readWrite – Geethika
* committx, aborttx, do\_commit\_abort – Snigdha
* set\_lock and documentation –Snigdha and Geethika

**Logical errors:**

1. At the start, we did not know how to compile the given code, so we used Cygwin instead of omega and tried to run it with g++ compiler, we encountered errors and warnings just for the given code, so we used omega instead.
2. Segmentation fault-When we run the test-files we always got this error.
3. The biggest task was set\_lock method as a lot was going on there. We had to check if the transaction was null and if null add from the hashtable, or find it and apply perform\_readWrite, we faced many difficulties in this area and how to deal with semaphores.
4. For version2, we faced difficulty while opening log file as it would return an error opening file message, so we had to change the logfile to logfilename in zgt\_tm.c.