

acm International Collegiate Programming Contest
sponsored by IBM

Problem 5: STM

Executable Program : Prog5.exe, Prog5.class
Source Program: Prog5.cpp, Prog5.java, Prog5.pas
Input: Prog5.in, Output: Standard Output

Efficient Developments ships a product which requires some database features. Since the company can't assume an existing DBM installed in the client system and their distribution must be as small as possible. Their solution is to create a Simple Transaction Manager.

This Simple Transaction Manager must be able to retain information through its executions and to avoid several client processes to access the same data at the same time. Client process will be identified, on the test file, as a three uppercase letter string followed by a colon, the command the client is requesting with its corresponding parameters, if any. The commands that can be issue to the STM are:

STARTSERVER
SHUTDOWN
CRASH
STARTTRANSACTION
COMMIT
ROLLBACK
READ Record
UPDATE Record Value
CREATEDATABASE Size

The database has at most 500 registers but it must support to read a four digits size number (such as 0500) in what can be thought of as a single table. All values and all record keys have the four digits, left zero padded numbers are used.

In order to access the server, it must be running. The server will support only one transaction per client at the same time, and all records accessed (read or write) while the transaction is open will be locked for the other clients. When an operation which requires a running server is performed against an stopped server (all except crash, createdatabase and startserver) the response: "ERROR: NO SERVER UP" is generated,

The database can only be created when the server is not running,
The server cannot be started while running, nor shutted down if it not running
The database must be started before it can be shutted down.
To shutdown a database there must be no pending transactions.

The server can be “crashed” at any time: it is stopped, regardless of its status or pending transactions.

When a transaction is committed, changes are made permanent. The information should exist between calls to the STM. A CREATEDATABASE statement will destroy all existing data and initialize all records to “0000”. A record update outside a transaction is permanent. When a Transaction is rolledback, all updates to records within that transaction are discarded.

The STM outputs to a file, the client who issued a command, and the command output:

Please see sample output files.

The following table shows the possible message the STM systems should produce.

<i>Command</i>	<i>Success</i>	<i>Error Message</i>	<i>Error Message</i>
STARTSERVER	OK	ERROR: SERVER UP	
SHUTDOWN	OK	ERROR: NO SERVER UP	ERROR: OPEN TRANSACTIONS
CRASH	OK		
STARTTRANSACTION	OK	ERROR: A TRANSACTION IS ALREADY OPEN	
COMMIT	OK	ERROR: NO OPEN TRANSACTION	
ROLLBACK	OK		
READ Record	Value	ERROR: RECORD IS LOCKED	
UPDATE Record Value	OK	ERROR: RECORD IS LOCKED	
CREATEDATABASE Size	Value	ERROR: SERVER UP	

NO Read/Update operation will try to access non existing records at database. The server is shutted, not shouted!!! (the word is misspelled all along)

SAMPLES

Input:

YYY: CREATEDATABASE 500
XXX: STARTSERVER
AAA: READ 0001
AAA: UPDATE 0001 1222
AAA: STARTTRANSACTION
AAA: UPDATE 0001 1500
BBB: STARTTRANSACTION
AAA: READ 0001
BBB: UPDATE 0315 0123
CCC: STARTTRANSACTION
DDD: UPDATE 0316 0456
CCC: STARTTRANSACTION
CCC: READ 0001
BBB: COMMIT
XXX: SHUTDOWN
CCC: UPDATE 0317 0789
CCC: COMMIT
XXX: CRASH
AAA: UPDATE 0001 1300
DDD: READ 0245

Output:

YYY: OK
XXX: OK
AAA: 0000
AAA: OK
AAA: OK
AAA: OK
BBB: OK
AAA: 1500
BBB: OK
CCC: OK
DDD: OK
CCC: ERROR: A TRANSACTION IS ALREADY OPEN
CCC: ERROR: RECORD IS LOCKED
BBB: OK
XXX: ERROR: OPEN TRANSACTIONS
CCC: OK
CCC: OK
XXX: OK
AAA: ERROR: NO SERVER UP
DDD: ERROR: NO SERVER UP

Input:

AAA: READ 0001
BBB: READ 0315
DDD: READ 0316
CCC: READ 0317
DDD: READ 0245

Output:

AAA: ERROR: NO SERVER UP
BBB: ERROR: NO SERVER UP
DDD: ERROR: NO SERVER UP
CCC: ERROR: NO SERVER UP
DDD: ERROR: NO SERVER UP

Input:

XXX: STARTSERVER
AAA: READ 0001
BBB: READ 0315
DDD: READ 0316
CCC: READ 0317
DDD: READ 0245

Output:

XXX: OK
AAA: 1222
BBB: 0123
DDD: 0456
CCC: 0789
DDD: 0000