

Implementação de um escalonador de transações para um SGBD multiusuário (módulo de controle de concorrência)

Objetivo: implementar, de forma simulada, um escalonador de transações que utilize o protocolo 2PL Strict.

Organização : o trabalho deve ser individual ou em duplas, e a escolha da linguagem de programação é livre.

Funcionamento geral da implementação:

As transações a serem executadas ou são informadas (digitadas suas operações em sequência – cadastro das transações) ou são escolhidas entre as que já estão armazenadas. Cada transação tem um identificador, e possui *start*, *commit* e operações de leitura (r) e/ou escrita (w) de dados.

Como é uma simulação, os dados são representados por variáveis do programa.

A seguir o simulador gera uma possível sequência de execução com operações entrelaçadas, a história inicial de execução (HI). Na sequência, essas operações da HI vão sendo enviadas ao escalonador que controla a execução, na ordem em que estão na HI, e o escalonador vai verificando se podem ser executadas nessa sequência ou não:

- para cada operação, verifica se o *lock* correspondente à operação pode ser concedido; se puder concede o *lock* e coloca a operação na história final de execução (HF); se não puder, coloca a operação em *delay* (e por consequência as demais operações da mesma transação também ficam em *delay*) e passa para a próxima operação prevista na HI. Antes de verificar cada nova operação prevista na HI, verifica se as operações em *delay* poder ser executadas (se puderem, executa).

Quando uma operação pode ser executada, ela é colocada na HF.

No final, a sequência de operações executadas está na HF (a ordem das operações das transações na HF pode ser diferente da ordem em que elas foram fornecidas na HI).

OBS: pode-se também permitir que seja fornecida uma HI previamente definida, para facilitar os testes com histórias conhecidas.

Observações:

- quando o protocolo utilizar *locks* (e *unlocks*): eles devem ser apresentados na HF (e os *delays* de operações controlados). A HI não tem *locks*, pois eles são colocados pelo escalonador;
- quando houver *deadlocks*: eles devem ser sinalizados (mostrar na tela), e o escalonador deve ter um critério para tratá-los (definir um critério e utilizá-lo para tratar);
- quando houver *abort* de transações: as operações de cada transação abortada devem ser retiradas da HF, e essas transações devem ser colocadas novamente em execução pelo simulador.

O banco de dados a ser utilizado pelas transações (dados para *read* e *write*) pode ser simulado por um conjunto de variáveis em memória (não precisa estar em arquivo), e é composto pelos dados (variáveis) utilizados nas transações.

Para entregar/apresentar (*upload* no webfólio correspondente):

Relatório: Fazer uma pequena descrição do trabalho realizado, descrevendo:

- protocolo escolhido e características de seu funcionamento (inclusive critério para *aborts* em caso de *deadlocks*)
- características implementadas e não implementadas
- como deve ser a entrada dos dados
- o que é gerado como saída de dados.
- estrutura de dados para armazenar: as transações, as histórias, o controle dos *locks*, o controle de execução e os dados

Implementação:

- cópia do código fonte
- executável (se for o caso)

O trabalho (execução e código-fonte) deverá ser apresentado à professora **até dia 17/06/2019**.