

# ACS Assignment 3

This assignment is due via Absalon on December 8, 23:59. This assignment consists of peer review of Assignment 2. As such, it should be solved using the same group structure as in Assignment 2.

Each group will be assigned the solutions of five other groups. A well-formed solution to this assignment is a ZIP file containing a set of peer reviews encoded in the XML templates provided, covering the work of the five groups under review. The templates must be filled after careful consideration of the instructions provided below; in particular, we expect that you consider the expectations for each question, and evaluate if these expectations have been met by the solutions of your colleagues.

This assignment will be passed upon submission of a well-formed hand-in ZIP file with your peer reviews. Note that all homework assignments have to be submitted via Absalon in electronic format. It is your responsibility to make sure in time that the upload of your files succeeds. Email or paper submissions will not be accepted.

## Learning Goals

This assignment targets the following learning goals:

- Evaluate the quality of arguments analyzing the serializability of transaction schedules, as well as the predicted outcomes of concurrency control protocols, e.g., variants of two-phase locking (2PL) or optimistic concurrency control.
- Provide feedback on the depth of arguments justifying the design of concurrency control mechanisms for a particular system in which operations must be guaranteed to be atomic, taking into special account the issues of correctness by equivalence to a 2PL variant and the consideration of aspects such as predicate reads, deadlocks, and the amount of concurrency achieved.
- Evaluate and provide feedback on implementations of concrete locking strategies in a modular service to guarantee the atomicity of operations in multithreaded executions.

## Peer Review of Exercises

For each of Questions 1 and 2 in Assignment 2, first consider the solution sketches provided in file `a2-exercise-solution-sketches.pdf`. Then, for each solution provided to each item in the questions, address the following evaluation criteria:

1. Is the solution provided both correct **and** includes a convincing short argument justifying it? [Yes/No] Note that simple statements of the solution without an accompanying short argument should be marked as No.
2. If the answer to the question above is No, state why you find the argument provided insufficient or incorrect. [Short free text, 1-2 sentences; feedback does not need to be extensive in this part since solution sketches are available]

## Peer Review of Programming Task

To craft your peer reviews of the programming task, we ask you to take as a starting point the answers provided to the questions in the original programming task description, and dive into the handed-in code to seek for supporting evidence. We detail the criteria that should be addressed for each of the original questions below:

1. **Question 1, Assignment 2: Description of implementation and tests.** Address the following criteria in your review:
  - (a) Does the solution contain a clear description of the approach used to achieve before-or-after atomicity? [Yes/No] A clear description is defined as one that characterizes the strategy followed in the code at a high level. Since the rationale for correctness and performance is provided in the subsequent questions, justification for the strategy used is not expected at this point.
  - (b) Provide comments on any aspects that you found unclear in the description of the approach to achieve before-or-after atomicity (i.e., the description either did not reflect what is in the code, or failed to provide explanation for what is in the code). [Free text]
  - (c) Does the code include implementations of Tests 1 and 2 of the assignment description as well as of at least another two test cases? [Yes/No]
  - (d) The solution should contain a clear description of the strategies employed to test correctness of the concurrent implementation, in particular for the verification that anomalies do not occur. For a description to be considered enough clear, it should relate the tests developed in the code to the notions of correctness established by serializability theory. In particular, two notions should be addressed:
    - i. Is it tested that the *final state* resulting from concurrent execution is correct in that it respects invariants designed in the test? [Yes/No]
    - ii. Is it tested that *intermediate reads* of the state *during concurrent execution* respect invariants designed in the test, so that it is arguable that these intermediate states could be found in an equivalent serial schedule? [Yes/No]

- (e) Provide comments on why you found that any portions of the test descriptions were unclear according to the criteria, and suggestions for improvement. [Free text]
2. **Question 2, Assignment 2: Correctness argument for locking protocol.** Address the following criteria in your review:
- (a) Is the variant of 2PL to which equivalence is being established in the correctness argument clearly identified (e.g., conservative, strict, conservative strict)? [Yes/No]
  - (b) Is there a sound argumentation of why the steps in the locking protocol designed correspond to steps taken in the chosen variant of 2PL? [Yes/No] NOTE: the steps should be clearly identified with reference to the code; “proof by obviousness” is *not* acceptable.
  - (c) Is there an argument for the need to take into account predicate reads, with appropriate justification? [Yes/No]
  - (d) Provide comments on why you found any aspects of the correctness argument provided to be lacking, incorrect, or in need of improvement. [Free text]
3. **Question 3, Assignment 2: Deadlocks.** Address the following criteria in your review:
- (a) Does the solution contain a correct argument for why deadlocks can occur in the locking protocol or why deadlocks cannot occur? [Yes/No] Note that the argument should be both sound and reflect what has been implemented in the code.
  - (b) Provide comments on how the argument provided with respect to deadlocks could be improved. [Free text]
4. **Question 4, Assignment 2: Reflections on scalability.** Address the following criteria in your review:
- (a) Is there a sound argument about the scalability limits of the concurrent implementation provided? [Yes/No] Note that a sound argument should describe how the locking protocol implemented would behave as more hardware resources (e.g., CPUs) are provided in a single machine. Also, it should clearly discuss if the protocol would be applicable to multiple machines if the resources of a single machine are exhausted.
  - (b) Provide comments on how the reflections on scalability could be improved. [Free text]
5. **Question 5, Assignment 2: Argumentation on decisions made for performance.** Address the following criteria in your review:

- (a) Is there a sound argument in the solution about the trade-off between locking overhead and concurrency? [Yes/No] A sound argument should clearly identify how much locking overhead is expected, and how the concurrency is limited by the locking protocol as more clients access the service. Note that the latter may require that assumptions be stated about the workload (e.g., distribution of client operations).
- (b) Provide comments on how the argument about performance trade-offs in the locking protocol could be improved. [Free text]