Exam topics for Concurrent Programming Languages.

Oleg Batrashev

May 29, 2009

These are the topics for the Concurrent Programming Languages course exam.

• For every topic there are several keywords or ideas that should be described in the answer but not limited to them. The more you know about the topic the better.

• I’ll divide 20 questions into 5 tickets with 4 questions each. Each question is worth 10 points, 40 in total for one ticket.

• Exam type is mixed: written + oral. I expect to see one/two paragraphs for each question with, possibly, some examples. I’ll check the answer and ask additional questions about the topic if needed.

• Short examples in any applicable language (even pseudocode) are welcome but no syntactic validity is necessary for the code, just to make sure you grasp the ideas.

Introduction

1. Definition(s) of parallel and concurrent, where and how parallelism is used.

2. Classifications of concurrency.

Declarative concurrency

3. Declarative programming: declarativeness, iterative and recursive computations, accumulators.

4. Purely functional (immutable) data structures: example structures, behavior with multiple threads.

5. Declarative concurrency: presence or lack of determinisms, evaluation orders, race conditions, exceptions.

Message-passing concurrency

7. Port concept, why we need it, equivalency with cells.
8. Port objects (agents), reasoning with agents, state diagrams.

Shared-state concurrency

10. Explicit state, programming with explicit state – limiting interleavings, equivalency with ports.
12. Confinments, active objects (actors).

Distributed programming

15. General ideas: network transparency, network awareness, openness, fault tolerance.
17. Mozart 1.4 fault tolerance model: fault stream, fault states for different entities.
18. Erlang fault tolerance model: linking, system process, EXIT signal handling.

Data parallel languages

19. Flat data parallelism: cache coherence, OpenMP.