

Threading and concurrency - Mopup

Radu Nicolescu
Department of Computer Science
University of Auckland

7 June 2013

① Threading Revision

② Threading Exam

③ Next

Outline

- ① Threading Revision
- ② Threading Exam
- ③ Next

Threading Revision

- Threading and concurrency
- Order over chaotic background
- Benefits and risks (race conditions, thread-safety, deadlocks)
- Fast growing area: software development tries to catch up with hardware developments (multi-cores, many-cores)
- Assignment: genetic programming, randomised algorithms
- Lesson: no free lunch, costs of using too many temporary objects, such as threads

Threading Revision

- Threading and concurrency
- Order over chaotic background
- Benefits and risks (race conditions, thread-safety, deadlocks)
- Fast growing area: software development tries to catch up with hardware developments (multi-cores, many-cores)
- Assignment: genetic programming, randomised algorithms
- Lesson: no free lunch, costs of using too many temporary objects, such as threads

Threading Revision

- Threading and concurrency
- Order over chaotic background
- Benefits and risks (race conditions, thread-safety, deadlocks)
- Fast growing area: software development tries to catch up with hardware developments (multi-cores, many-cores)
- Assignment: genetic programming, randomised algorithms
- Lesson: no free lunch, costs of using too many temporary objects, such as threads

Threading Revision

- Threading and concurrency
- Order over chaotic background
- Benefits and risks (race conditions, thread-safety, deadlocks)
- Fast growing area: software development tries to catch up with hardware developments (multi-cores, many-cores)
- Assignment: genetic programming, randomised algorithms
- Lesson: no free lunch, costs of using too many temporary objects, such as threads

Threading Revision

- Threading and concurrency
- Order over chaotic background
- Benefits and risks (race conditions, thread-safety, deadlocks)
- Fast growing area: software development tries to catch up with hardware developments (multi-cores, many-cores)
- Assignment: genetic programming, randomised algorithms
- Lesson: no free lunch, costs of using too many temporary objects, such as threads

Threading Revision

- Threading and concurrency
- Order over chaotic background
- Benefits and risks (race conditions, thread-safety, deadlocks)
- Fast growing area: software development tries to catch up with hardware developments (multi-cores, many-cores)
- Assignment: genetic programming, randomised algorithms
- Lesson: no free lunch, costs of using too many temporary objects, such as threads

Outline

- ① Threading Revision
- ② Threading Exam
- ③ Next

Threading Exam

- 30 out of 100 exam marks
- 14 MCQ: 1-2 marks each
- 3 MCQ + short answer: 3-4 marks each
- If you believe that a question has either NO or MULTIPLE correct answers, select the ONE you believe is MOST LIKELY to be the intended answer (and try to explain why).
- Please be very careful at NEGATIVE questions, of the genre “which of the following items does NOT ...” or “which of the following answers is FALSE”

Threading Exam

- 30 out of 100 exam marks
- 14 MCQ: 1-2 marks each
- 3 MCQ + short answer: 3-4 marks each
- If you believe that a question has either NO or MULTIPLE correct answers, select the ONE you believe is MOST LIKELY to be the intended answer (and try to explain why).
- Please be very careful at NEGATIVE questions, of the genre “which of the following items does NOT ...” or “which of the following answers is FALSE”

Threading Exam

- 30 out of 100 exam marks
- 14 MCQ: 1-2 marks each
- 3 MCQ + short answer: 3-4 marks each
- If you believe that a question has either NO or MULTIPLE correct answers, select the ONE you believe is MOST LIKELY to be the intended answer (and try to explain why).
- Please be very careful at NEGATIVE questions, of the genre “which of the following items does NOT ...” or “which of the following answers is FALSE”

Threading Exam

- 30 out of 100 exam marks
- 14 MCQ: 1-2 marks each
- 3 MCQ + short answer: 3-4 marks each
- If you believe that a question has either NO or MULTIPLE correct answers, select the ONE you believe is MOST LIKELY to be the intended answer (and try to explain why).
- Please be very careful at NEGATIVE questions, of the genre “which of the following items does NOT ...” or “which of the following answers is FALSE”

Threading Exam

- 30 out of 100 exam marks
- 14 MCQ: 1-2 marks each
- 3 MCQ + short answer: 3-4 marks each
- If you believe that a question has either NO or MULTIPLE correct answers, select the ONE you believe is MOST LIKELY to be the intended answer (and try to explain why).
- Please be very careful at NEGATIVE questions, of the genre “which of the following items does NOT ...” or “which of the following answers is FALSE”

Outline

- ① Threading Revision
- ② Threading Exam
- ③ Next

Next

- Research projects (Radu)
 - Bio-inspired parallel and distributed models of computations: formal models, algorithms, graph theory, efficient parallel implementations on a variety of targets
- Further teaching (Radu)
 - 280: C#, databases
 - 335: functional programming layer, based on list comprehensions, that uniformly integrates objects, databases, XML docs, web objects, parallel computing
 - 734: advanced functional programming, F#, with applications to parallel computing

Next

- Research projects (Radu)
 - Bio-inspired parallel and distributed models of computations: formal models, algorithms, graph theory, efficient parallel implementations on a variety of targets
- Further teaching (Radu)
 - 280: C#, databases
 - 335: functional programming layer, based on list comprehensions, that uniformly integrates objects, databases, XML docs, web objects, parallel computing
 - 734: advanced functional programming, F#, with applications to parallel computing

Next

- Research projects (Radu)
 - Bio-inspired parallel and distributed models of computations: formal models, algorithms, graph theory, efficient parallel implementations on a variety of targets
- Further teaching (Radu)
 - 280: C#, databases
 - 335: functional programming layer, based on list comprehensions, that uniformly integrates objects, databases, XML docs, web objects, parallel computing
 - 734: advanced functional programming, F#, with applications to parallel computing

Next

- Research projects (Radu)
 - Bio-inspired parallel and distributed models of computations: formal models, algorithms, graph theory, efficient parallel implementations on a variety of targets
- Further teaching (Radu)
 - 280: C#, databases
 - 335: functional programming layer, based on list comprehensions, that uniformly integrates objects, databases, XML docs, web objects, parallel computing
 - 734: advanced functional programming, F#, with applications to parallel computing