Thread Synchronization II

Søren Hansen <sha@ase.au.dk>
V1.0

Introduction

Using protection inadvertently presents some challenges, namely *when to use*, *how to use* and what might go run if used without care. These, rather abstract, notation are the topics for this lecture.

Content and reflection

Themes

- Deadlock
  - Definition[4][1]
  - Dining philosophers problem[5]
    Read chandra/misra only as inspirational
- General
  - Thread Models[3, chap. 4]
    Other kinds of thread models(!)
  - Thread Safety[3, chap. 5]
  - Rules for Multithreaded Programming[3, chap. 6]
- Priority Inversion[6]
  Note: Read for understanding bounded and unbounded from a basic point of view. The article is quite extensive and the required knowledge is what is presented in the slides.
- Buffer case - Producer/Consumer[7][2]
  Implemented using semaphores

Questions

- Pitfalls
  - Initialising semaphores - how and with what number
  - What to lock and when to lock
  - Locking / unlocking - missing one spells what?
- Deadlock
  - Which conditions need to be present for a deadlock to occur
  - What is the problem behind the *Dining philosophers problem*
  - Which rule can you do something about, and what could you do to solve this particular problem.
- Priority inversion
  - What actually makes this problem
  - What are the two different strategies to solve the problem
  - In what way do they solve the problem and what is the consequence of which when doing so.

Material

Slides

[1] S. Hansen, *Thread synchronization ii*, Slides - see course repos.

[2] ——, *Thread synchronization i*, Slides - see course repos.

Local repository

[3] P. C. Chapin, *Pthread tutorial*, Tutorial, See https://redmine-server.ase.au.dk/courses/projects/i3isu/repository, 2008.

Online

[4] E. al. (). "Deadlock," [Online]. Available: `https://en.wikipedia.org/wiki/Deadlock`.

[5] ——, (). "Dining philosophers problem," [Online]. Available: `https://en.wikipedia.org/wiki/Dining_philosophers_problem`.

[6] B. R. Kyle Renwick. (). "How to use priority inheritance," [Online]. Available: `http://www.embedded.com/design/configurable-systems/4024970/How-to-use-priority-inheritance`.

[7] E. al. (). "Producer–consumer problem." Wikipedia Article, [Online]. Available: `https://en.wikipedia.org/wiki/Producer%2dconsumer_problem`.