PADTAD 2012

Curriculum for software development with concurrency
- Jeremy S. Bradbury, João M. Lourenço, Ok-Kyoon Ha, In-Bon Kuh, Yong-Kee Jun
Concurrent Programming

At the undergraduate level, where is concurrency taught in your university?

- 2nd year Java Programming course/Programming Language & Environments course – (1 week at end of course)
- 3rd year software quality course – (< 1 week)
- 3rd year Unix System Programming – (2 weeks) concurrency with signal handler programming
- 2nd/3rd year Operating System course – (2-4 weeks) semaphores, locks, producer-consumer from systems perspective
- 4th year Distributed Systems course – only indirectly as it related to distributed
Concurrency – Education

• Should concurrency be taught at the undergraduate level?

  • YES!

• If so when?
  • Should continue to be integrated into 2\textsuperscript{nd} and 3\textsuperscript{rd} year programming, OS, Intro. to SE courses
  • Should have dedicated concurrency course in 4\textsuperscript{th} year (or 3\textsuperscript{rd} year)
Concurrency – Education

• **Does language matter in a dedicated concurrency course?**
  • Could use different languages (partially language agnostic)
    • Java, C#, C++, Scala
  • **Threads are a must!** Maybe incremental enhance programs from threads on single machine up to HPC and cloud computing
Possible topics for a dedicated course

• Why concurrency? Motivation and introduction
• Multi-core CPU programming as well as GPU (at least a little)
  • Constructs: synchronized, semaphores, actors, transactions, etc.
  • Advanced topics: cloud computing, OpenCL (heterogeneous)
• Basics in problem solving with parallel – using constructs above
  • Parallel algorithms: Searching, Sorting, Map/reduce
• Quality Control and Assurance
  • Verification & Validation – testing