This is a sample syllabus only. The instructor may make changes to the syllabus in future courses.

Multiprocessor Programming

Course Syllabus - CS 692/792 - Spring 2015

Class: MWF 2:30pm - 3:20pm in EB127

Last Updated: November 23, 2015

Instructor: Peter Pirkelbauer
Email: pirkelbauer@uab.edu
Office: CH 143
Office Hours: W 10:00am - 11:00am and by appointment
Information: Course schedule and announcements will be published on Canvas (check often!).

1 Prerequisite

Course in parallel / concurrent / distributed programming, or approval by the instructor.

2 Description

This course examines synchronization in concurrent systems, available atomic primitives, non-blocking programming techniques, lock-/ wait-freedom, transactional memory, and memory models. The application of these techniques to the development of scalable data structures for multi-core architectures will also be discussed.

3 Course Philosophy

Writing correct and efficient concurrent code requires a comprehensive understanding of modern hard- and software systems, including system architectures, programming languages, compilers, and run-time systems. CS692/792 is a research oriented course on the development of concurrent software. The course will examine foundations in hard- and software and discuss their use for the development of scalable data structures on multi-core architectures and beyond. A large portion of the class material will draw from research papers. Students will demonstrate mastery in the field by developing a research project. To attain a good implementation, the project will require from students the combination and application of multiple concepts discussed in class.

4 Learning Objectives

Students with a basic knowledge in compilers, architecture, and concurrent/parallel/distributed systems will become familiar with atomic primitives offered by modern instruction set architectures (ISA). They will understand the application of these atomic primitives to the construction of classes of concurrent algorithms, such as blocking, termination-safe, obstruction-free, lock-free, and wait-free algorithms. Students will be able to distinguish memory consistency models offered by modern hardware designs, such as x86, PowerPC, Alpha, and ARM processors and know about their practical implications. Students will become familiar with memory consistency models defined for common programming languages, such as C++ and Java. Students will be able to identify common problems in concurrent and in non-blocking codes and know how to avoid them. They will explore methods to prove the correctness of (non-blocking) concurrent codes. Students will know how to apply transactional memory for the development of concurrent systems. As alternative to atomic primitives, students will evaluate transactional memory as a means for developing scalable systems.

Undergraduate students will use this knowledge to reimplement existing data-structures and compare their efficiency. Graduate students are encouraged to utilize the material to tackle a research problem that is relevant to the topic of this course. This can be either the novel implementation of data-structures, proofs of correctness claims of existing implementations, or similar projects.

5 Course Material

Textbook: Maurice Herlihy and Nir Shavit, The Art of Multiprocessor Programming, Revised First Edition, Morgan and Kaufmann, 2012. ISBN13 978-0123973375. The textbook will provide the foundation for this course.
Research papers: Some classes will be taught based on research papers. References and other required readings will be added to Canvas.

6 Grading

A major component of the class grade will e determined by the class project. The class project will be carried out by teams of two (a different number of team members requires approval by the instructor). The project will be graded from several perspectives. This includes a final project report, a project presentation, and biweekly progress presentations. Also, class participation and assignments will be graded. There will be no exam in this class.

- *Class Project* (200 points) Grading the project will assess the difficulty of the topic, the approach to solve the problem, and its implementation. Graduate students are encouraged to tackle a relevant research problem related to the course. Undergraduate teams can also choose to experiment with existing approaches (based on the research literature).

- *Project Report:* (150 points) The project report is a 8 to 15 pages write up on your project (using ACM sigplanconf.cls or similar style). Grading the report will be based on content, grammar, style, quality of writing and relevance of presentation. Note, whether or not the outcome of the project is publishable has no impact on the grade. Title and a short outline are due on January 23rd, the full paper is due on April 20th.

- *Project Presentation:* (150 points) At the end of the semester each project team will give a 25 minute presentation (20 minutes presentation + 5 minutes for questions and answers). Each team member has to speak for an about roughly the same amount of time. The presentation slots will be assigned on a first come, first serve basis.

- *Lightning Talks:* (150 points) Biweekly each project team will give a 4-6 minute presentation on the progress of their project. The speakers of the team alternate; each talk counts towards the grade of the presenter but not for the other team members. Lightning talks start on February 11th.

- *Participation:* (150 points) The class will discuss recent research papers on non-blocking programming. Active participation in class exercises, discussions, and lectures will be graded. Attendance alone will not be sufficient to receive full credit.
- *Assignments:* (200 points) There will be five or more homework assignments in the semester. The best four assignments will count towards the grade.

IMPORTANT: Papers may be submitted to Turnitin for comparison against the most comprehensive digital repository of potentially plagiarizable material. Your work may be stored in Turnitin's database and used as reference for future comparisons.

Final grade: The final grade is determined by the sum of the individual grading categories as outlined above.

- A: 1000 — 900
- B: 899 — 800
- C: 799 — 700
- D: 699 — 600

6.1 Late Submission

All assignments are due as indicated on Canvas. Any assignment turned in after this deadline is considered late and may not be graded.

6.2 Class Attendance

Attendance of student project presentations at the end of the semester is mandatory. Attendance of other classes is strongly recommended. A large portion of the grade depends on your active class participation. Just showing up to class will not be enough for a good participation grade.
Students auditing this course are expected to follow the same attendance guidelines.

6.3 Academic Honesty Policy

Any student who violates the university's academic honesty policy will be reported for academic discipline. All university and department policies related to students are included here by implication.

7 Disability Support Services

If you are registered with Disability Support Services (DSS), please make an appointment with me as soon as possible to discuss accommodations that may be necessary. If you have a disability but have not contacted DSS, please call 934-4205, visit DSS at the 9th Ave. Office Building, 1701 9th Ave. South, or contact them through the web `http://www.uab.edu/dss`.

8 Add/Drop Policy

- A student can **add** or **drop** the course through dates as specified in UAB's academmic calendar.

- A student can **withdraw** with a "W" by dates as specified as specified in UAB's academmic calendar. (Undergraduate)
- For more details about add/drop policies check with Registration/Academic Records.

9 Communication

E-mail: Students will be required to use *their UAB email address* blazerid@uab.edu. New students must login and configure their email addresses. For more details on obtaining blazerid and configuring email please see: http://www.uab.edu/blazerid. Key email communications will be made using this address. You must regularly log in to Canvas, since emails will also be sent there and they do not forward. Assignments and information will appear only there.
See the course's *Canvas* site — `http://www.uab.edu/online/canvas` — regularly for updates.

Dept. of Computer and Information Sciences
The University of Alabama at Birmingham
115A Campbell Hall, 1300 University Boulevard
Birmingham, Alabama 35294-1170

10 Tentative Class Schedule

Date	Day	Topic	Assignment
Jan 5	M	*canceled*	Reading: HS ch. 1
Jan 7	W	Introduction	
Jan 9	F	A Lock-free Stack	
Jan 12	M	Mutual Exclusion	Reading: HS ch. 2
Jan 14	W		
Jan 16	F	A Lock-free Queue	
Jan 19	M	Martin Luther King Holiday	
Jan 21	W	Consistency of Concurrent Objects	Reading: HS ch. 3, pp 45-61 (excl. 3.8)
Jan 23	F	Memory Management in C++	*due:* Project Proposal
Jan 26	M	Concurrent Objects (Formalism)	
Jan 28	W	Atomic Snapshots	Reading: HS ch. 4, pp 87-93
Jan 30	F	Memory Management in C++	
Feb 2	M	*Project Work*	
Feb 4	W	Atomic Operations	Reading: HS ch. 5
Feb 6	F	Hardware Memory Models	Paul McKinney: Memory Ordering in Modern Microprocessors, web published, 2007
Feb 9	M	Consensus	Reading: HS ch. 6
Feb 11	W		
Feb 13	F	Memory Consistency Java & C++	HS: ch 3, pp 61-64 Boehm and Adve: Foundations of the C++ concurrency memory model, PLDI 2008.
Feb 16	M		
Feb 18	W		
Feb 20	F		
Feb 23	M	Spin Locks and Contention	Reading: HS ch. 7
Feb 25	W		
Feb 27	F		
Mar 2	M	Concurrent Stacks and Queues	Reading: HS ch. 10 + 11
Mar 4	W		
Mar 6	F	Concurrent Skiplist	Reading: HS ch. 14
Mar 9	M		
Mar 11	W		
Mar 13	F	Model Checker	Reading: Demsky et al.: CDSChecker, OOPSLA 2013.
Mar 16	M		
Mar 18	W		
Mar 20	F		
Mar 23	M	*Spring Break*	
Mar 25	W	*Spring Break*	
Mar 27	F	*Spring Break*	

Mar 30	M	*Project Work*	
Apr 1	W	Multi-compare and swap	Harris et al.: A practical multi-word compare-and-swap operation, DISC 2002.
Apr 3	F	Concurrent Hash tables	Reading: HS ch. 13
Apr 6	M		
Apr 8	W	Transactional Memory	Reading: HS ch. 18
Apr 10	F		
Apr 13	M	Hardware Transactions	
Apr 15	W	Barries	Reading: HS ch 19
Apr 17	F	Counting	Reading: HS ch 14
Apr 20	W		*due:* Project Report