Course	CSI 4360/5360, Winter 2018 Concurrent and Multicore Programming Dept. of Computer Science and Information, Oakland University
Credit	4
Lecture	Sections 14452, MW 3:30 – 5:17 PM, Pawley Hall 308
Instructor	Xiaotong Lin xlin@oakland.edu Office Hours: TR 12:00-1:00 PM, and by appointment, at DH129

Textbooks and Course materials:

There are not any required textbooks for this course. The following is a list of recommended **open source** books.

1) Introduction to Parallel Computing, 2nd Edition
   By A. Grama, A. Gupta, G. Karypis, and V. Kumar *Addison-Wesley, 2003*
   http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.726.694&rep=rep1&type=pdf
2) CUDA By Example: An Introduction to General-Purpose GPU programming
   By J. Sanders and E. Kandrot
   http://www.itp.cas.cn/kxjs/jzytz/pxyjz/201306/P020130624362235915673.pdf
3) Programming on Parallel Machines: GPU, Multicore, Clusters and More
   By Norm Matloff, University of California, Davis
   http://heather.cs.ucdavis.edu/~matloff/158/PLN/ParProcBook.pdf

Course Description

This course will focus on concepts, theory, design and implementation of concurrent programs for multi-core computers, multicore programming methodologies. Topics covered include mutual exclusion, memory model and thread-based parallelism, fork join framework, locks, parallel control flow, concurrent data structures.

Requirements

In order to perform well in this course, the followings are helpful: Experiences in C programming and Linux environment; basic knowledge of computer architecture, Operating systems, data structures, and compilers; skills in reasoning and theoretical analysis.

Objectives

By successfully completing this course, students should be able to (letters in parenthesis indicate ABET student outcomes):

1. Describe benefits and applications of concurrent and parallel programming. (a,b,g)
2. Explain key concepts in parallel computer architectures, e.g. shared memory system, distributed system, NUMA and cache coherence. (a,b,c,g,h)
3. Understand principles for concurrent program design, e.g. decomposition of works, task and data parallelism, processor mapping, mutual exclusion, locks. (a,b,c,g, h, I, j, k)
4. Write parallel program using OpenMP, Cilkplus, CUDA, MPI programming models. (a,b,c,d,f,I,j,k)
5. Perform analysis of parallel program problems. (a,b,c,d,f,i,j,k)

Resources

Documentations of the SECS computing facilities: http://www.secs.oakland.edu/docs/pdf

Grade Assessment

Grade weights: Homework 45%, Project 15%, Midterms 30%, and Quizzes 10%

Numeric course grades (4.0-0.0) are determined with formula:
Grade* = *Score/16 -2, where *Score* is in 100 scale

Homework
We will have three assignments finished individually. The assignments will be due at 11:55pm on the specified dates, unless instructed differently. Without informing the instructor in advance, *no late work will be accepted*.

Project
This group-oriented project aims at applying the knowledge learnt in class to solve a challenging parallel computing problem. Group members need to collaborate in the process of investigating the problem, researching related work, finding a solution, implementing the design, analyzing the results, and formally documenting and presenting your work.

Midterms
We will have two closed-book midterms, each of which weighs 15% of the course grade. Refer to the Tentative Class Schedule for the exam dates.

Quizzes
Occasionally throughout the semester, we will have some popup quizzes without announcements in advance. They may take place anytime during a lecture. Thus, your class attendance is highly recommended.

Advice for performing well in this class

1. Attend the class and actively engage in class activities
2. Keep up with the assignments, since many of the concepts build upon each other.
3. Review the assignments when assigned (even if you don't have time to work on them right then). This way you can plan out your week and get your questions answered early. Don't wait until the last minute to work on an assignment at home.
4. Read ahead in the book. Many of the questions you encounter in the homework and exams can be found in the reading.
5. Check Moodle website of this session at https://moodle.oakland.edu/moodle often for updates. This website will contain notes, assignments, supplementary materials, assignment due dates, exam dates, etc.
6. Class announcements and reminders through Moodle will go to your OU email. If you don't check OU email very often, forward it to your primary email account.
7. Last but not least, if you have trouble understanding a concept, please contact me right away. Best way to catch me is by email (xlin@oakland.edu).

Behavioral Contract
Please mute your cell-phone during lectures. If you need to use your laptop or a computer in the classroom, please sit at the back of the room, in order to minimize the distraction you cause to fellow classmates.

Academic Conduct
It is assumed that all work throughout the term is your own. Discussion of assignments is permitted but copying of assignments or parts of assignments is not. Oakland University Academic Conduct Policy can be found at http://wwwp.oakland.edu/studentcodeofconduct/

Tentative Class Schedule

Week	Topic	Note
1 **(Jan 3)**	Intro to parallel computing	
2 **(Jan 8, 10)**	Review C and Linux OpenMP	
3 **(Jan 15, 17)**	OpenMP	**No class** on MLK day (01/15)
4 **(Jan 22, 24)**	Parallel Algorithm Design	
5 **(Jan 29, 31)**	Parallel Algorithm Design	
6 **(Feb 5, 7)**	Measure and Analysis	
7 **(Feb 12, 14)**	Measure and Analysis	Midterm-I on 02/14
8 (**Feb 19, 21)**	No classes	Winter Recess
9 **(Feb 26, 28)**	PThread	
10 **(Mar 5, 7)**	Parallel Architecture	
11 **(Mar 12, 14)**	GPU/CUDA	
12 **(Mar 19, 21)**	GPU/CUDA	
13 **(Mar 26, 28)**	MPI	
14 **(Apr 2, 4)**	MPI	
15 **(Apr 9, 11)**	Parallel Algorithms-applications	Midterm-II on 04/11
16 **(Apr 16)**	Project Presentations	
Final Exams	**No Final Exam**	