CSCI 3366 (Introduction to Parallel and Distributed Processing), Spring 2011

Syllabus

1 Course description

This course is an introduction to parallel and distributed programming, including both the concepts and their application. Course content will include discussions of different types of parallel machines and machine models, the design and analysis of parallel algorithms, and the development of parallel programs.

The objectives of this course include, but are not limited to, the following:

- Learning fundamental concepts of parallel programming.
- Learning parallel algorithm design.
- Learning the basics of parallel machine structure.
- Programming using message-passing (e.g., using MPI).
- Programming using threads (e.g., using OpenMP and/or Java).

2 Basic information

Class meeting times and location

- TR 3:35pm – 4:50pm, HAS 340

Prerequisites

- CSCI 2320, or consent of instructor

Instructor contact information

- Dr. Berna Massingill
- Office: HAS 201L
- Office phone: (210) 999-8138
- E-mail: `bmassing@cs.trinity.edu`

Office hours

Scheduled office hours for this semester are as follows:

- Monday 12:30pm – 1:30pm
- Tuesday noon – 12:30pm, 2pm – 3:30pm
- Wednesday noon – 4pm
- Thursday noon – 12:30pm, 2pm – 3:30pm

These times are subject to change; a current schedule will be available on my Web page.

If I'm not in my office, I should be somewhere in the building (perhaps in one of the labs helping another student), and there will often be a note on my door saying where to find me.

Some office hours will be held in one of the classrooms/labs (times to be announced soon). These are "open lab" times, during which I'll be in one of the department's labs, prepared to answer questions. The intent is that students can use these times to work on assignments with someone available to help with any questions or problems.

In addition to scheduled office hours, you're welcome to drop by and see if I'm in my office and free to talk, or you can make an appointment by calling me or sending me e-mail.

E-mail is almost always a good way to reach me; I normally check it fairly often and reply promptly.

3 Course materials

Textbook

- Timothy G. Mattson, Beverly A. Sanders, and Berna L. Massingill. *Patterns for Parallel Programming.* Addison-Wesley, 2004.

Web page

Most course-related information (this syllabus, homework and reading assignments, etc.) will be made available via the Web. The course Web page is a starting point for Web-accessible course material; you can find it linked from my home page (`http://www.cs.trinity.edu/~bmassing`) or directly at `http://www.cs.trinity.edu/~bmassing/Classes/CS3366_2011spring/HTML`.

Other references

There are many, many books on parallel computing. Below is a selection of books I have found interesting or useful over the years.

- Gregory R. Andrews. *Multithreaded, Parallel, and Distributed Programming.* Addison-Wesley, 2000. A more theory-oriented treatment.
- M. Ben-Ari. *Principles of Concurrent and Distributed Programming.* Prentice Hall, 1990. A more theory-oriented treatment.
- Rohit Chandra, Leonardo Dagum, Dave Kohr, Dror Maydan, Jeff McDonald, and Ramesh Menon. *Parallel Programming in OpenMP.* Morgan Kaufmann, 2000.
- K. Mani Chandy and Jayadev Misra. *Parallel Program Design: A Foundation.* Addison Wesley, 1989. A much more theory-oriented treatment, co-authored by my thesis advisor.
- Jack Dongarra, Ian Foster, Geoffrey Fox, Ken Kennedy, Andy White, Linda Torczon, and William Gropp, editors. *The Sourcebook of Parallel Computing.* Morgan Kaufmann, 2002.
- Ian Foster. *Designing and Building Parallel Programs.* Addison Wesley, 1995.
- Brian Goetz, Tim Peierls, Joshua Bloch, Joseph Bowbeer, David Holmes, and Doug Lea. *Java Concurrency in Practice.* Addison-Wesley, 2006.
- William Gropp, Ewing Lusk, and Anthony Skjellum. *Using MPI: Portable Parallel Programming with the Message-Passing Interface.* The MIT Press, second edition, 1999.
- William Gropp and Marc Snir. *MPI: The Complete Reference.* MIT Press, second edition, 1998.
- Doug Lea. *Concurrent Programming in Java: Design Principles and Patterns.* Prentice Hall, second edition, 1999.

- Peter Pacheco. *Parallel Programming with MPI.* Morgan Kaufmann, 1996.
- Michael J. Quinn. *Parallel Programming in C with MPI and OpenMP.* McGraw-Hill, 2004. The textbook I would use if mine didn't exist.
- Barry Wilkinson and Michael Allen. *Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers.* Prentice Hall, 1999. A book Dr. Eggen and I have used in the past as a textbook for this course.

4 Course requirements

Grading

Grades in this course will be determined by the results of several homework assignments, a project, and class participation, weighted as follows.

Component	**Maximum points**
Homework	about 150
Project	80
Class participation	20

Numeric grades will be calculated as a simple percentage, by dividing total points earned on the above components by total points possible. These numeric grades will then be converted to letter grades based on a curve, but in no case will the resulting letter grades be worse than students would receive based on the following scheme.

Numeric grade	**Letter grade**
90 – 100	A-/A
80 – 89	B-/B/B+
70 – 79	C-/C/C+
60 – 69	D/D+
0 – 59	F

Homework assignments

Several homework assignments will be required for successful completion of this course; many will involve programming. Detailed requirements, including due dates and times, will be provided as part of each assignment. Programming problems will be coded using suitable parallel languages or libraries (C with MPI or OpenMP, or Java) as specified in individual assignments.

Project

As part of the course, students must also complete a significant project approved by the instructor and present it to the class; students may work individually or in groups of two. Detailed requirements for the project will be described separately and will include program code, a short written report, and a presentation to the class.

Notice that although there are no exams in this course, we will use the time scheduled for a final (May 4 at 8:30am) for project presentations. Please plan accordingly (i.e., avoid scheduling anything else for that time).

Attendance

Regular class attendance is strongly encouraged; class participation grades will be based largely on attendance.

E-mail

Course-related announcements will sometimes be made by sending e-mail to the Trinity e-mail addresses of all registered students. Students are strongly encouraged to read mail sent to their Trinity addresses frequently.

Late and missed work

Unless otherwise stated for a particular assignment, homework will be accepted up to one class period late, *but no more*, at a penalty of 10 percent off per working day. This penalty may be waived or additional time allowed *at the instructor's discretion* in cases of illness or conflict with a university-sponsored activity or religious holiday.

If you have unusual circumstances (as we all sometimes do), please discuss these with me as far in advance as possible.

Academic integrity at Trinity

All students are covered by the Trinity University Honor Code, which prohibits dishonesty in academic work.

The Code asserts that the academic community is based on honesty and trust, and defines specific violations as well as the procedure to determine if a violation has occurred. The Code also covers the process of hearings for alleged violations and the various sanctions applied for specific violations. The Code also provides for an appeal process.

The Code is implemented by the Academic Honor Council. Under the Code, a faculty member will (or a student may) report an alleged violation to the Academic Honor Council. It is the task of the Council to collect the pertinent evidence, adjudicate, and assign a sanction within certain guidelines if a violation has been verified.

Students who are under the Honor Code are required to pledge all written work that is submitted for a grade: "On my honor, I have neither given nor received any unauthorized assistance on this work" and their signature. The pledge may be abbreviated "pledged" with a signature.

The specifics of the Honor Code, its underlying philosophy, and the norms for sanctioning can all be found on the Academic Honor Council website, accessed through the Trinity Homepage, or directly here[1].

Collaboration and academic integrity in this course

Unless otherwise specified, all work submitted for a grade (homework assignments and projects) must represent the student's own individual effort. Unless otherwise stated, all submitted work will be considered pledged work.

Discussion of homework assignments and course material among students is encouraged, but not to the point where detailed answers are being written collectively. Graded papers and sample solutions from previous years (for this course, homeworks) are off limits. Answers that are identical beyond coincidence (either to another student's work or to a sample solution from a previous year)

[1] `http://www.trinity.edu/departments/academic_affairs/honor_code/`

will be considered to be in violation of the Honor Code, and *will result in appropriate action.* You are responsible for the security of your work, both electronic and hard copy.

Introduction to File Processing (3) Prerequisite: CSCI 115. Definition of file components, access methods, and file operations. Algorithms for efficient implementation of data structures; characteristics of bulk storage media for mainframe and microcomputers.Â Multiprocessing vs. multicomputers. Shared memory vs. message-passing, problem solving, and parallel programming. Architectural trends. CSCI 177. Spring 2011. Department of Computer Science, University of Colorado at Boulder.Â Distributed Systems: Concepts and Design, by George Coulouris, Jean Dollimore, and Tim Kindberg, Addison Wesley, 4th edition, 2005. (CDK). Distributed Algorithms (The Morgan Kaufmann Series in Data Management Systems) by Nancy A. Lynch, 1996. class Web site: See the Moodle class Web page at http://moodle.cs.colorado.edu. Assignments, lecture slides, and announcements can be found there. The moodle has a variety of useful features, including a forum for posting questions. Each student should establish an account on the moodle and then subscribe to our class on the moodle using the special enrol