CSCI 3366
(Introduction to Parallel and Distributed Processing)
Spring 2001
Syllabus

1 Course description

This course is an introduction to parallel and distributed processing, including both the theory and the application of parallel-processing concepts. 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 parallel processing concepts.
- Learning parallel algorithm design.
- Learning parallel machine structure.
- Mapping theoretical problem solutions to physical hardware.
- Programming using message-passing (e.g., using MPI).
- Programming using threads (e.g., using Java).

2 Basic information

Class meeting times and location:

TR 12:45pm – 2:00pm, Halsell 228

Prerequisites:

CSCI 2320, or consent of instructor

Instructor:

Dr. Berna Massingill

Office: Halsell 201L

Office phone: (210) 999-8138

Web page: `http://www.cs.trinity.edu/~bmassing`

E-mail: `bmassing@cs.trinity.edu`

Office hours: MW 12:30pm – 3:30pm, TR 2:00pm – 4:00pm, and by appointment

3 Course materials

Textbook:

Barry Wilkinson and Michael Allen. *Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers.* Prentice Hall, 1999.

Web page:

Most course-related information (this syllabus, homework and reading assignments, etc.) will be made available via the World Wide Web. The home page for the course is not only a starting point for Web-accessible course material but will also be used for course-related announcements. Please plan to check it frequently. You can find it linked from my home page (`http://www.cs.trinity.edu/~bmassing`) or directly at `http://www.cs.trinity.edu/~bmassing/CS3366_2001spring/`.

Other references:

- Selim G. Akl. *The Design and Analysis of Parallel Algorithms.* Prentice Hall, 1989.
- Kenneth A. Berman and Jerome L. Paul. *Fundamentals of Sequential and Parallel Algorithms.* PWS, 1997.
- Steven Brawer. *Introduction to Parallel Programming.* Academic Press, 1989.
- K. Mani Chandy and Jayadev Misra. *Parallel Program Design: A Foundation.* Addison Wesley, 1989.
- Bruno Codenotti and Mauro Leoncini. *Introduction to Parallel Processing.* Addison Wesley, 1993.
- T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms.* McGraw Hill, sixth edition, 1992.
- Michel Cosnard and Denis Trystram. *Parallel Algorithms and Architectures.* PWS, 1995.
- Ian Foster. *Designing and Building Parallel Programs.* Addison Wesley, 1995.
- A. Geist, A. Beguelin, J. Dongarra, W. Jiang, R. Manchek, and V. Sunderam. *PVM: Parallel Virtual Machine. A Users' Guide and Tutorial for Networked Parallel Computing.* The MIT Press, 1994.
- William Gropp, Steven Huss-Lederman, Andrew Lumsdaine, Ewing Lusk, Bill Nitzberg, William Saphir, and Marc Snir. *MPI: The Complete Reference. Volume 2, The MPI-2 Extensions.* The MIT Press, 1998.
- William Gropp, Ewing Lusk, and Anthony Skjellum. *Using MPI: Portable Parallel Programming with the Message-Passing Interface.* The MIT Press, 1999. Fifth Printing.
- Marc Snir, Steve W. Otto, Steven Huss-Lederman, David W. Walker, and Jack Dongarra. *MPI: The Complete Reference. Volume 1, The MPI-1 Core.* The MIT Press, second edition, 1998.

4 Course requirements

Grading:

Grades in this course will be determined by the results of two major exams, several homework assignments, and a course project. The two exams will be equally weighted; weights for homework assignments and the course project will vary with the difficulty of the assignment. Numeric grades will be calculated as a simple percentage, by dividing points earned (on homework assignments and exams) by 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
80 – 89	B
70 – 79	C
60 – 69	D
0 – 59	F

Exams:

Exams are comprehensive but will emphasize the most recent material. They are scheduled as follows. Please plan accordingly. Note that there is no final exam as such, but the date and time reserved for the course's final will be used for project presentations.

- Exam 1: February 15, in class. *Changed to:* February 22, in class.
- Exam 2: April 12, in class. *Changed to:* April 26, in class.
- Final exam period (to be used for project presentations): May 8, 2:00pm.

Homework assignments:

Several homework assignments, including programming problems, will be required for successful completion of this course. 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 (e.g., Java or MPI) as specified in individual assignments.

Course 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, made during the time scheduled for the course's final exam.

Attendance:

Regular class attendance is strongly encouraged.

Late and missed work:

Exams and project presentations can be made up only in cases of documented conflict with a university-sponsored activity or documented medical emergency.

Homework will normally 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.

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

Collaboration and academic integrity:

Unless otherwise specified, all work submitted for a grade (homework assignments and exams) must represent the student's own individual effort. Discussion of homework assignments among students is encouraged, but not to the point where detailed answers are being written collectively. Answers that are identical beyond coincidence are in violation of Trinity's Academic Integrity Policy and will result in disciplinary action, including, but not limited to, a failing grade on that assignment for all parties involved. You are responsible for the security of your work, both electronic and hard copy.