SIMD and Multicore Programming

The Art Institute of Portland℠

Course Description

VGP393C

Summer 2008, 3 credits

Wednesdays, 6:00PM - 9:45PM

Room #214

In this course students will learn the fundmentals of designing and implementing parallel algorithms on multi-core, multi-threaded, and SIMD computer architectures. This includes the working knowledge of synchronization primitives, use of multi-threading programming interfaces, and use of SIMD processor extensions.

By the end of the course, students will be able to:

- Understand terminology and issues related to parallel programming.
- Understand and implement multi-threaded programs using the Win32 / MFC threading APIs.
- Understand and implement SIMD accelerated algorithms using the SSE, SSE2, and SSE3 processor extensions.

The complete, up to date, course syllabus is also available on-line at the course website (http://people.freedesktop.org/~idr/2008Q3-VGP393/). The syllabus is available as both HTML and PDF (http://people.freedesktop.org/~idr/2008Q3-VGP393/SU08_VGP393C_A.pdf).

Prerequisite

Successful completion of CS312 or consent of instructor is required.

This course is programming intensive. A strong background in C or C++ programming is required. Familiarity with object oriented programming principles will be very helpful but is not strictly required.

Texts

Required text:

Akhter, Shameem; Roberts, Jason. *Multi-Core Programming: Increasing Performance through Software Multi-threading*, Intel Press, April 2006. ISBN 0-9764832-4-6.

Required Materials

In addition to paper and writing utensils, each student will need a removable storage device. The storage device will be used to both bring documents and sample code home from class and bring homework completed assignments to class. The storage requirements should be minimal, so a small USB flash-drive (256MB) should be sufficient.

Grading

Each student's grade in this course will be primarily based on a total of five single-week programming assignments and one four-week programming project. Each student will also be required to read an academic paper and present a summary of that paper to the class. The remainder of the student's grade will be based on bi-weekly quizes and a final exam.

Programming assignments will be graded first and foremost on whether or not correct output is produced. The remaining points are based on the style of the program. This includes, but is not limited to, algorithm selection, code formatting, and naming conventions. A detailed rubric will be provided with each assignment.

Programming Assignments		
In-class presentation		20 pts.
Homework programming assignments		50 pts.
Term project		50 pts.
	Subtotal	120 (63%)
Tests		
In-class quizzes		20 pts.
Final Exam		50 pts.
	Subtotal	70 pts. (37%)
	Total	190 pts. (100%)

Some assignments *may* carry extra-credit opportunities, but they will be infrequent.

Grading Scale

A	=	93% and above
A-	=	90%-92%
B+	=	87%-89%
B	=	83%-86%
B-	=	80%-82%
C+	=	77%-79%
C	=	73%-76%
C-	=	70%-72%
D+	=	67%-69%

D = 60%-66%

Late Work

I do not accept late work. If you miss a deadline, you will not earn the points for that activity. There are no make-up opportunities. If you are unable to attend class on the due date for a assignment, please submit it by e-mail *before* class.

Attendance and Participation

If you are not in class for an in-class exercise, you cannot earn those points. If you miss an entire class, you are responsible for obtaining copies of handouts and other classroom materials from your classmates.

AiPD Policies

Lab Policies

Leave food and drink outside the class. Disciplinary action will be taken toward any student found using the equipment in an inappropriate manner, taking cell phone calls or surfing the web. Disruptive, disrespectful or rude behavior will not be tolerated.

Plagiarism

Presenting the writings, images or paraphrased ideas of another as ones own, is strictly prohibited at the Art Institute of Portland. Properly documented excerpts from others works, when they are limited to an appropriate amount of the total length of a student's paper, are permissible when used to support a researched argument.

Students with Disabilities

It is AiPD policy not to discriminate against qualified students with a documented disability in its educational programs, activities or services. If you have a disability-related need for adjustments or other accommodations in this class, contact the Disability Services Coordinator.

Amber Perrin
Disabilities Services Coordinator
The Art Institute of Portland
1122 NW Davis Street
Portland, OR 97209-2911

503-382-4836
<aperrin@aii.edu>

Course Calendar

Week 1 (July 16th)

Lecture notes. (20080716 - Intro.pdf)

- Course road-map
- Types of parallel computers
  - SISD / superscalar
  - SIMD
    - Vector processors
    - General purpose CPU extensions
  - MIMD
    - SMP
    - NUMA
    - SMT
    - CMP
- Multi-threading terminology
- Homework assignments:
  - Read *Multi-Core Programming*, chapter 1 and chapter 2.

Week 2 (July 23rd)

Lecture notes. (20080723 - Synchronization.pdf)

- Synchronization
  - Critical sections
  - Deadlock
  - Synchronization primitives
    - Semaphores
    - Locks

    - Condition variables

- Win32 / MFC threading API, part 1
    - Creating threads
    - Destroying threads
    - Events
    - Semaphores
    - Mutexes
    - Critical sections
    - Compiling / linking multi-threaded programs
- Homework assignments:
    - Read *Multi-Core Programming*, chapter 4 and chapter 5 up to the section “Atomic Operations” on page 91.
    - Programming assignment #1: Multi-threaded “Hello, world!”. (20080723_Assignment.pdf) Due 7/30. 10 points.

Week 3 (July 30th)

Lecture notes. (20080730 - Program decomposition.pdf)

- Quiz #1
- Finding Concurrency
    - Program decompositions
    - Dependency analysis
    - Design evaluation
- Homework assignments:
    - Read *Multi-Core Programming*, chapter 3.
    - Programming assignment #2: Fractal generator. (20080730_Assignment.pdf) Base code. (mandelbrot.zip) Due 8/20. 30 points.

Week 4 (August 6th)

Lecture notes. (20080806 - Algorithm Structure.pdf)

- Algorithm structure patterns
    - Task Parallelism

- Divide and Conquer
- Geometric Decomposition
- Recursive Data
- Pipeline
- Event-Based Coordination

Week 5 (August 13th)

No class this week. Prof. Romanick will be at SIGGRAPH.

Week 6.1 (August 20th)

Lecture notes. (20080820 - Supporting Structures.pdf)

- Quiz #2
- Supporting structure patterns
  - SPMD
  - Master / worker
  - Loop parallelism
  - Fork / join
  - Shared data
  - Shared queue
  - Distributed array

Week 6.2 (August 22nd)

Lecture notes. (20080822 - Atomic Operations.pdf)

- Atomic operations
- Lockless algorithms
- Win32 / MFC threading API, part 2
  - Thread pools
  - Thread priority
  - Processor affinity
  - Thread-local storage

Week 7 (August 27th)

Lecture notes. (20080827 - Threading Problems.pdf)

- Common Multi-threading Problems
  - Deadlock / livelock
  - Thread-safe libraries
  - Cache abuse / memory bandwidth
- Homework assignments:
  - Read *Multi-Core Programming*, chapter 7.
  - Programming assignment #3: Parallel game tree search. (20080827_Assignment.pdf) Due 9/10. 30 points. Sample code (othello.zip)

Week 8 (September 3rd)

No class this week. Prof. Romanick will be at the X Developer's Summit.

Week 9.1 (September 10th)

Lecture notes. (20080910 - SIMD Extensions.pdf)

- Quiz #3
- SIMD
  - Overview of common SIMD extensions
  - Data organization for SIMD
  - SSE, SSE2, SSE3
  - Future of SIMD
- Homework assignments:
  - Programming assignment #4: SIMD point classification (20080910_Assignment.pdf). Due 9/17. 30 points.

Week 9.2 (September 12th)

- Intel Threaded Building Blocks

Week 10 (September 17rd)

Final exam. 5:30PM - 7:30PM. **Do not be late today!**