Syllabus for Computer Architecture and Parallel Programming, CS 2420

Instructor: Dr. Mark Fienup, Computer Science Department
Office Location: 313 ITT
Office Phone: 319-273-5918
Email Address: fienup@cs.uni.edu
Office Hours: M: 8-11:45, 1:10-3,  T: 11-11:45, 1:10-2,  W: 1:10-2,  Th:11-11:45, 1:10-2,  F: 8-10

Course Information
Meeting Time & Location:
Course meets Tuesday and Thursday from 9:30 AM to 10:45 AM in ITT 328

Course Description:
Computer architecture of uniprocessor and multiprocessor systems with an emphasis on parallel programming to achieve high performance. Topics include processor design (pipelining and superscalar), memory hierarchy, interconnection networks, performance metrics, parallel program design, and parallel programming tools for multi-core systems, clusters, and graphics processing units (GPUs)

Prerequisites: Introduction to Computing (CS 1510) and Computer Organization (CS 1410)
The ability to program in a high-level programming language (e.g., Python, Java, C/C++, Ada, etc.) is necessary to succeed in this course. Successful completion of your Introduction to Computing course should provide you with sufficient programming ability. Knowledge of uniprocessor computer organization including components (CPU, memory, I/O devices, system bus, etc.), fetch-decode-execute cycle, and some understanding of assembly language programming are needed from your Computer Organization course.

Purpose of the course:
The purpose of this course is to enable you to write software that can utilize the computational power of current and future computer architectures. Your current skills only allow you to write single-thread programs which utilize only a single processor core. This might be okay if the computer only has 2 or 4 cores, but as the number of cores per processor increases your future employer will want you to do better. Already computers have programmable Graphics Processing Units (GPU, i.e., graphics cards) with thousands of processors, and computers are networked together into clusters, we’ll also learn how to write programs that leverage these technologies to vastly increase the computational power of your programs.

Course Organization:
This course is taught as a "flipped classroom." That means you’ll have textbook reading, video to watch, quiz to take, and questions to answer before each class. The video will cover the "easier" material for the class so we can spend in-class time focusing on the more challenging material.

The course is organized into five units:

1. **Uniprocessor computer architecture and C programming** – Before you can write efficient parallel programs, you must first learn how to design and write efficient uniprocessor programs. Understanding the computer architecture (i.e., pipelining and superscalar processor design and memory hierarchy) are key in achieving high performance. As you learn about computer architecture, you will also be learning to program in C which will be used to write sequential and parallel programs throughout the course. (Weeks 1 to 3)

2. **Parallel hardware and parallel software design** – Before you can design and write efficient parallel programs, you must also understand the parallel hardware on which your parallel programs will execute. As you learn about parallel hardware, you will also be learning general parallel program design techniques and patterns. (Weeks 4 and 5)

3. **Shared-memory programming with pthreads** – For a relatively small number (< 64) of processors (e.g., current multi-core computers), shared-memory design is common. A parallel program runs on multiple processors/cores as "light-weight" (i.e., low overhead points of execution in the same program) threads. These threads communicate data and synchronize tasks by reading and writing data in the shared memory. You will practice your parallel program design skills by writing several C program that utilize the pthread module to create and coordinate threads running on multiple processor cores. (Weeks 6 to 9)

4. **Distributed-memory programming with MPI** – As the number of processors/cores increase, they all contend for memory-bandwidth. Since memory holds their running programs and data, memory becomes a bottleneck (von Neumann bottleneck). Thus, all large parallel computers are distributed-memory machines, i.e., a collection of networked, “stand-alone” computers each with their own local memory. A parallel program running across multiple computers needs to communicate data and synchronize tasks using the network. You will practice your parallel program design skills by writing several C program that utilizing MPI (Message-Passing Interface) functions which is the standard distributed-memory programming tool. (Weeks 10 to 11)

5. **General-purpose GPU programming with CUDA** – Driven by 3D computer games, graphics accelerator cards (i.e., GPU—Graphics Processing Unit) where developed with thousands of processors to perform mathematically intense real-time 3D graphical rendering. To leverage the computational power of GPUs for general purpose (i.e., non-graphics) computing tasks, NVIDIA modified their GPU architecture to create the CUDA Architecture about 2007. You will practice your parallel program design skills by writing several C program that utilizing CUDA extensions to execution on massively parallel GPU. (Weeks 12 to 15)

Textbooks and Course Packet
**Required:** *An Introduction to Parallel Programming*, Peter S. Pacheco, 2011, 1st Edition, Morgan Kaufmann Publishers,
Print Book ISBN: 9780123742605 (~ $80)
eBook ISBN: 9780080921440

*Computer Architecture (CS 2420 - Fienup) Course Packet* – available at UNI Bookstore (1009 W. 23rd St., Cedar Falls, IA 319-273-2665)

Course Learning Objectives

Learning Objective #1: Explain the operation of uniprocessor computer components including the processor (pipelined and superscalar) and memory hierarchy (cache and virtual memory).

Learning Objective #2: Demonstrate an understanding of uniprocessor computer architecture by designing and writing C programs that make efficient use of the processor and memory hierarchy.

Learning Objective #3: Explain the operation of parallel hardware including cache-coherence and mutexes on shared-memory machines, and interconnect performance (bisection bandwidth, bandwidth and latency) characteristics on distributed-memory machines.

Learning Objective #4: Demonstrate an understanding of parallel hardware and general parallel program design techniques and patterns by producing efficient parallel program designs to minimize parallel program overhead.

Learning Objective #5: Demonstrate an understanding of shared-memory machines by designing and writing C programs using the pthread module that make efficient use of multiple cores.

Learning Objective #6: Demonstrate an understanding of distributed-memory machines by designing and writing C programs using the MPI module that make efficient use of multiple networked processors.

Learning Objective #7: Demonstrate an understanding of GPU programming by designing and writing CUDA programs that make efficient use of the GPU processing power.

Course Requirements

Instructional Methods / Activities / Assessments

This course consists of the following activities and assessments to assist you in achieving the course and unit objectives. For consistency and planning purposes, each week is structured the same and consists of the following:

<table>
<thead>
<tr>
<th>Activity/Assessment</th>
<th>Due Date and Time</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Textbook Reading</td>
<td>Before the Tuesday Video</td>
<td>You can do these in any order that you find helpful. Often the video will provide more depth or further explanation of the textbook’s coverage.</td>
</tr>
<tr>
<td>“Tuesday” Video Online Quiz</td>
<td>Tuesday at 9:30 AM</td>
<td>Automatically graded eLearning quiz—typically 5 multiple choice questions</td>
</tr>
<tr>
<td>Lab Video</td>
<td>Mid-week/After Tuesday video</td>
<td>Zip together programs and answers to written questions. Graded by instructor using the Lab Rubric.</td>
</tr>
<tr>
<td>Lab .zip submission</td>
<td>Saturday at 11 PM</td>
<td></td>
</tr>
<tr>
<td>Textbook Reading 2</td>
<td>Before the Thursday Video</td>
<td>You can do these in any order that you find helpful. Often the video will provide more depth or further explanation of the textbook’s coverage.</td>
</tr>
<tr>
<td>“Thursday” Video Online Quiz</td>
<td>Thursday at 9:30 AM</td>
<td>Automatically graded eLearning quiz—typically 5 multiple choice questions</td>
</tr>
<tr>
<td>Weekly Homework Assignment</td>
<td>The following Wednesday at 5 PM (HWs 7 - 10 are due in two-weeks)</td>
<td>The first 5 HWs will consist of written and problem based questions. The remaining HWs will be programs and graded using the Homework Rubric.</td>
</tr>
</tbody>
</table>

Grading

There are a total of 590 points possible across all assessment activities. The following table shows a breakdown of these points.

<table>
<thead>
<tr>
<th>Assessment Activity</th>
<th>Number of Assessments</th>
<th>Points per Assessment</th>
<th>Total for Assessment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weekly Labs</td>
<td>13</td>
<td>10 points each</td>
<td>130 points</td>
</tr>
<tr>
<td>Homework Assignments</td>
<td>10 (HWs 7 - 10 are two weeks)</td>
<td>20 points each</td>
<td>200 points</td>
</tr>
<tr>
<td>Online Video Quizzes</td>
<td>28</td>
<td>5 points each</td>
<td>140 points</td>
</tr>
<tr>
<td>Unit Quizzes: Units 1 to 4</td>
<td>15 points each</td>
<td>60 points</td>
<td>60 points</td>
</tr>
<tr>
<td>Final Exam – includes Unit 5</td>
<td>60 points</td>
<td>60 points</td>
<td>60 points</td>
</tr>
<tr>
<td>Total Points</td>
<td></td>
<td></td>
<td>590 points</td>
</tr>
</tbody>
</table>

Your course grade will be based by the percentage of these 590 points that you accumulate. Grades will be assigned based on straight percentages off the top student score. If the top student’s percentage is 94%, then the grading scale will be 94-84% A’s, 83.9-74% B’s, 73.9-64% C’s, 63.9-54% D’s and below F’s.

Technology Requirements:

The following information has been provided to assist you in preparing to use technology successfully in this course. You may need to install free software on your computer. You will need the following:

- Internet access/connection – high speed recommended (not dial-up)
- Email – mostly to ask questions asynchronously
- Word Processor (e.g., Open-office, MS Word, etc.)
- SSH and telnet client (free options for Windows: PuTTY from http://www.putty.org/; MAC or Linux users can use: ssh userName@computerName in a terminal window)
- Secure file transfer client (free options for Windows: FileZilla from https://filezilla-project.org/; MAC or Linux users can use: scp -r userName@computerName:remoteDir localDir
Communication and Support:
Even though this is a Face-to-Face course, you will need to communicate with the instructor and fellow students using several forms:

- Face-to-Face meeting during office hours (listed above) are my preferred form of communication on technical course material. You can come to my office or arrange a Google Hangout.
- Email – email should be used if you have personal concerns or questions. I will typically respond within a couple of hours, but maybe as longer at night or on weekends. If several students ask the same question, I will post an answer to the Frequently Asked Questions (FAQ) eLearning Discussion Board.
- eLearning Announcements – periodically I will post eLearning Announcements to alert you to important course related information

Grading response time
Students will have immediate feedback on online quiz grades. For labs and homework assignments, I will make every attempt to grade and post those grades within a week after their submission time has expired. Sometimes, however, other university work commitments may delay the grading process.

Technical Support
If you experience any technical problem do not contact the instructor. Please contact the Office of Continuing and Distance Education at 319-273-7740 or cesp-consult@uni.edu. Support hours are available Tuesday – Thursday 8am – 9pm and Thursday 8am – 5pm.

Course and University Policies:

Late Assignment Policy
Any type of assignment not submitted by its submission time is considered late. Late assignments will be accepted, but will incur a 20% penalty.

Honor Code
Copying from other students is expressly forbidden. Doing so on assignments will be penalized every time it is discovered. The penalty can vary from zero credit for the copied items (first offense) up to a failing grade for the course. If an assignment makes you realize you don’t understand the material, ask questions designed to improve your understanding, not ones designed to discover how another student solved the assignment. The solutions to assignments should be individual, original work unless otherwise specified. Remember: discussing assignments is good. Copying code or question answers is cheating.

Any substantive contribution to your assignment solution by another person or taken from a publication (or the web) should be properly acknowledged in writing. Failure to do so is plagiarism and will necessitate disciplinary action. Additionally, assisting or collaborating on cheating is cheating. Cheating can result in failing the course and/or more severe disciplinary actions. You are responsible for being familiar with complete set of University Academic Ethics Policies [http://www.uni.edu/pres/policies/301.shtml].

Office of Student Disability Services
The Americans with Disabilities Act of 1990 (ADA) provides protection from illegal discrimination for qualified individuals with disabilities. Students requesting instructional accommodations due to disabilities must arrange for such accommodation through the Office of Student Disability Services (SDS). The SDS is located at: 103 Student Health Center, the phone number is: (319) 273-2677, and the website is http://www.uni.edu/sds/.

UNI Academic Ethics/Discipline Policy
Students at the University of Northern Iowa are required to observe the commonly accepted standards of academic honesty and integrity. [http://www.uni.edu/policies/301]

UNI Student Code of Conduct Policy
The university's student conduct code maintains the principles of respect, honesty, and responsibility to create a safe, healthy environment for members of the campus community while preserving an educational process that is consistent with the mission of the University. [http://www.uni.edu/policies/302]