CS 330 – Introduction to Real-Time Systems

General Course Info

Term:	Winter 2020
Department:	CS
Course Number:	330
Time:	3a (MW 11:10am – 12:20pm, F 12:00pm-1:00pm)
Location:	CMC 301
Website:	https://www.cs.carleton.edu/faculty/tamert/cs330-w20
Piazza:	https://piazza.com/carleton/winter2020/cs330

Instructor Info

Name:	Tanya Amert, Visiting Instructor
Office:	CMC 318
Email:	tamert@carleton.edu
Web:	https://www.cs.carleton.edu/faculty/tamert
Office Hours:	MW 3:30pm-4:30pm, T 1pm-3pm, Th 6pm-8pm, or by appt.

Textbooks and Resources

Required Textbook:
For this course, we will use *Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications*, by Giorgio C. Buttazzo (ISBN: 1461406757). You should be able to access the PDF online for free through the campus library (https://apps.carleton.edu/campus/library/).

All lecture and homework materials will be available on the course website: https://www.cs.carleton.edu/faculty/tamert/cs330-w20. In addition, programming assignments will be submitted through Moodle.

We are using Piazza as an online forum. You can ask questions about any of the course material on Piazza, but you must refrain from posting solutions to the homework problems.

Course Description

How can we prove that dynamic cruise control will brake quickly enough if traffic suddenly stops? How must a system coordinate processes to detect pedestrians and other vehicles to ensure fair sharing of computing resources? In real-time systems, we explore scheduling questions like these, which require provable guarantees of timing constraints for applications including autonomous vehicles.

This course will start by considering such questions for uniprocessor machines, both when programs have static priorities and when priorities can change over time. We will then explore challenges introduced by modern computers with multiple processors. We will consider both theoretical and practical perspectives.

Topics: uniprocessor scheduling, multicore scheduling, synchronization, parallel task scheduling, mixed criticality

Prerequisites

Prerequisites: Computer Science 201 and Computer Science 202 (Mathematics 236 will be accepted in lieu of Computer Science 202).

Target Audience

This course surveys the scheduling and synchronization algorithms needed to design a real-time system. We will utilize the proof techniques from CS 202 or MATH 236 to analyze and verify the correctness and runtime complexity of the algorithms discussed. We will also build on some of the data structures introduced in CS 201. In addition, written homework and programming assignments reinforce the concepts through further analysis and implementation.

Given that CS 201 is a prerequisite, students are expected to have a working knowledge of a programming language; for this course, we will use Python.

Goals and Key Learning Objectives

Upon completing this course, students should be comfortable analyzing and implementing a variety of algorithms for scheduling and synchronization. In addition, students should be able to determine requirements for a given problem, and be able to choose the proper algorithm to solve the problem.

Course Requirements

The homework assignments will be a combination of written homework questions to give students theoretical practice, as well as programming assignments for practical application. The programming assignments will build up a codebase for experimenting with different scheduling algorithms and synchronization protocols.

For a final project, students will choose a scheduling algorithm or synchronization protocol to add to the existing experimentation code base. They will implement the chosen algorithm, provide unit tests, and perform experiments to compare it to other approaches. There will be a project writeup and a project presentation near the end of the term.

During the term, there will be two midterm exams, as well as a final exam. All exams are cumulative.

Key Dates

First day of class: Monday, January 6th
Last day of class: Wednesday, March 11th

No class: Monday, February 10th

Midterm exam #1: Wednesday, January 29th
Midterm exam #2: Wednesday, February 26th
Final exam: Monday, March 16th, 3:30 p.m. – 6:00 p.m.

Attendance and Participation

Although attendance will not be taken in lecture, you are expected to attend all lectures. Participation will also be measured through the use of Piazza, both in asking and helping answer questions. There will be at least 5 in-class pop quizzes, used to gauge the pace and clarity of the material. These will either be anonymous or used to count towards participation; scores will not impact students' grades.

Course Policies

In class, you are expected to maintain proper etiquette. This includes arriving on time, not having conversations during lecture, and most importantly not having your laptop/phone/newspaper/etc. out during lecture.

Each midterm exam will be 70 minutes in duration, given during a class period. The course final will be 2.5 hours in duration, and will be given in compliance with Carleton College final exam regulations and according to the Carleton College Final Exam calendar.

Grading Criteria

Homework Assignments	36%
Project	12%
Midterm #1	12%
Midterm #2	12%
Final Exam	24%
Participation	4%

Assignments (written and programming) will be graded both for correctness and style. For written proofs, good style implies that you give enough information to complete the proof, and are not excessively verbose. For programs, this means having well-documented (through comments and/or good variable/function names) and easily readable code.

Late Policy

Each student starts the term with three late days. These can be used on any written or programming assignment for any reason, no explanation necessary, and at most two can be used on any given assignment. Note that one late day counts as a calendar day, not as 2-3 days until the next class session.

All written assignments will be due at the start of class on the day listed. If turned in after the start of class (including at the end of class), it will count as using a late day. Programming assignments will be due to be submitted to Moodle by the time listed on the assignment.

If you have no more late days, late assignments will not be accepted unless there are extenuating circumstances that you have discussed (in person, or over email and received an acknowledging reply) before the due date.

There will be optional problems on some assignments that count towards gaining additional late days. At most three* additional late days can be earned throughout the term. If you earn these extra late days but do not use them, they will turn into 1.0%* extra credit each at the end of the term (the original three late days do not count for this). ***UPDATE: Originally, four additional late days could be earned, and they would convert to 0.5% each. This was changed 3/4/20.**

Academic Integrity

For written homework assignments, you are allowed to discuss your general approaches with other students, but your final solution must be your own. On the first page of your homework submission, you must list the names of everyone you talked with and include a signed academic integrity statement.

For programming assignments, you are allowed to work with up to one other student. You may look at your partner's code and help with debugging, but you are expected to type up all of your code yourself. If you worked with someone else, you must each include each other's names in your readme.txt file submitted with your code. You may also discuss general approaches with others, and their names must be listed in your readme.txt file as well. See the assignment instructions for more details.

Exams are closed-book, closed-note. You will be allowed an increasing number of "cheat sheets" for the exams: 1 for the first mid-term, 2 for the second, and 3 sheets for the final. Your cheat sheets can be hand-written or typed on 8.5"x11" paper, double-sided, and you are expected to create your own. You are allowed to bring a magnifying glass.

If any student is suspected to have violated the academic integrity policy, a report will immediately be made to the Academic Standing Committee, as described at https://apps.carleton.edu/campus/doc/integrity. Ask the instructor if you are unsure about what constitutes acceptable collaboration.

Course Schedule

Weeks 1-3	Uniprocessor scheduling algorithms
Week 4	Synchronization on uniprocessor systems, midterm #1
Weeks 5-6	Multicore scheduling algorithms
Week 7	Multicore scheduling cont'd, multicore synchronization protocols
Week 8	Multicore synchronization cont'd, midterm #2, parallel tasks
Week 9	Parallel task scheduling cont'd^, project presentations
Week 10	Mixed criticality^, course recap + final review

^ Some topics may be skipped, shrunk, or replaced depending on time constraints.

Homework assignments will typically be due every other week, with varying proportions given to written and programming questions.

We will not meet for class on Monday February 10th due to Midterm Break.

Inclusion

I strive to create an inclusive and respectful classroom that values diversity. Our individual differences enrich and enhance our understanding of one another and of the world around us. This class welcomes the perspectives of all ethnicities, genders, religions, ages, sexual orientations, disabilities, socioeconomic backgrounds, regions, and nationalities.

Accommodations for Students with Disabilities

Carleton College is committed to providing equitable access to learning opportunities for all students. The Disability Services office (Henry House, 107 Union Street) is the campus office that collaborates with students who have disabilities to provide and/or arrange reasonable accommodations. If you have, or think you may have, a disability (e.g., mental health, attentional, learning, autism spectrum disorders, chronic health, traumatic brain injury and concussions, vision, hearing, mobility, or speech impairments), please contact disability@carleton.edu or call Sam Thayer ('10), Accessibility Specialist (x4464) or Chris Dallager, Director of Disability Services (x5250) to arrange a confidential discussion regarding equitable access and reasonable accommodations.

Disclaimer

The instructor reserves to right to make changes to the syllabus, including homework and project due dates and test dates. These changes will be announced as early as possible.