Syllabus CSCE 496/896-003: Real-Time Systems

Spring 2017

Updated: March 9, 2017

Instructor: Justin Bradley 261 Avery Hall
Office hours: by appointment
Email: justin.bradley@unl.edu
Phone: (402) 472-5072

Course Overview

This course introduces students to the design of real-time computing systems and corresponding theory. Real-time systems applications typically involve safety or mission critical systems where timing guarantees must be maintained. Real-time task design, dependability, communication, power and energy awareness, scheduling, and real-time control are some of the topics that will be covered. The course will be biased toward real-time system design of controlled systems, particularly robotics. The course will be based on hands-on labs over the course of the semester which consist of using the C programming language and a real-time operating system for the development of the RTS for a small mobile robot. Students will investigate timing guarantees, worst-case execution times of tasks, shared resources, task development, and guaranteed performance.

Class Details

Class: TR 09:30 - 10:45 AVH-110
Final exam: 10:00 to 12:00 Tuesday, May 2, 2017

Class website: `http://cse.unl.edu/~jbradley/courses/2017_spring/CSCE496-896-RTS/`

Note that I will run this course from UNL's new learning management system - Canvas - rather than Blackboard. So please use that.

Course Requirements

Text: There is no required textbook
However, here are a few other books that may be helpful for you. I have all of these so if you feel they would be useful let me know.

- Lee, Edward Ashford, and Sanjit Arunkumar Seshia. *Introduction to embedded systems: A cyber-physical systems approach.* Lee & Seshia, 2011.
- Alur, Rajeev. *Principles of Cyber-Physical Systems.* MIT Press, 2015.
- Wolf, Marilyn. *High-Performance Embedded Computing: Applications in Cyber-Physical Systems and Mobile Computing.* Elsevier, 2014.
- Simon, David E. *An Embedded Software Primer.* Addison-Wesley Professional, 1999
- Krishna, C. Mani. *Real-Time Systems.* John Wiley & Sons, Inc., 1999.
- Liu, Jane. *Real-Time Systems.* Prentice Hall, 2000

Prerequisites: I find that students have a hard time gauging their preparedness for interdisciplinary courses (yes this will be interdisciplinary) based only on the prerequisites given. As a result, I will break it down in terms of level of preparedness:

- Unprepared:
  - No knowledge of or courses in theoretical computation, operating systems, computer architecture, robotics, embedded systems, basic programming, algorithms, control theory
  - If this is you won't be allowed to stay in the course
- Borderline prepared:
  - One course from: basic programming, computer architecture, operator systems, embedded systems, or robotics
  - I will let you take the course but you should be prepared for some confusion requiring additional research and self-learning on your part
- Prepared:
  - Multiple courses from: robotics, programming, computer architecture, networking, control theory, operating systems, or related
  - You should be fairly comfortable in the course building on your robotics/computing foundation

Grading

This course will require active, in-class participation as part of group discussions, as well as participation in the labs. There will be approximately 10 labs moving you through the development of a RTS for a small mobile robot. There will then be a final exam in which you will need to demonstrate that your robot can complete an obstacle course within a certain amount of time.

Graduate Students

Graduate students must complete the labs and homework by themselves.

Undergraduate Students

Undergraduate students **may** work in teams of up to two for all assignments (including the final). If the class situation (i.e. odd number of undergraduate students, or some undergrads want to work on their own) requires any teams of three undergraduate students, that team will be required to do any bonus questions in each lab, as well as the bonus 11th lab.

Grade Breakdown

- labs - 70%
  - each lab is worth $^{70}/_{6} =\sim 11.67\%$
- homework - 15%
- in-class participation - 5%
  - class attendance. Attending...

    - lectures, prepares you to apply concepts from the course to contribute and complete the labs, particularly if you're on a team,
    - labs, lets me address specific lab issues and answer corresponding questions.
  - on-time. ***Please*** be on time. If you're consistently late I will dock you participation points relative to how often I observe you coming in late.
  - NOTE: 5% is enough to drop you two grades. As an example, if you have an A with 94% but get a 0 for participation, you would drop to B+ with 89%. So this score matters!
- final exam - 10%
  - the final exam will be handed out on Tuesday on the last week of class. This gives you time to prepare. The final exam will approximately equivalent to a lab but will include a demonstration of your robot accomplishing objectives at the final exam time.

Grades will then be assigned using the following table:

A+: [98-100]	A: [93-98)	A-: [90-93)
B+: [88-90)	B: [83-88)	B-: [80-83)
C+: [78-80)	C: [73-78)	C-: [70-73)
D+: [68-70)	D: [63-68)	D-: [60-63)
F: [0-60)		

Policies

Class Policies

1. Attendance at any individual lecture/lab is not absolutely mandatory, and it's fine to miss a class from time to time (but please let me know). However, you are responsible for anything that transpires during class, and the "participation points" are a subjective measure based on my impression of your participation. So you want to make sure my impression is that you are participating.
2. Exchange of ideas and techniques is **highly** encouraged but **your work must be your own**. Because we live in the age of GitHub and stackoverflow, learning from examples is understandable. But copying code directly is plagiarism and an academic violation and if I see it, I will turn you in for cheating. If you get help on an algorithm from the web that's fine, but you should write the code yourself, and put a comment in your code of where you found the example. Besides, you don't want to copy some code and introduce some nasty timing bug because a shared resource wasn't handled properly in the copied code!
3. All homework assignments must be typed unless otherwise specified. If you are not proficient with either LaTeX, Microsoft Word, or some other word processor or typesetting language now is your chance to learn as this will be a necessary component of your professional career.[1] If writing is difficult for you, or you are not proficient with English the UNL Writing Center (`http://www.unl.edu/writing/home`) may be able to help you. If I can't understand what you write I will dock you points. This is the subjective nature of writing and communicating.
4. Late work policy: no late work will be accepted. That is, you will get a zero if it is late. This is a real-time system course where we discuss the safety-critical nature of ensuring all deadlines are met. Ergo, all deadlines for all labs, homeworks, etc. are hard-deadlines. All assignments must be turned in prior to class on the day indicated, unless otherwise noted.
5. Students have one week from time of grade posting to challenge a grade.

[1] Personally, I use LaTeX, so if that's your platform and you need technical help I can offer it. If you use Microsoft Word I will be of no use to you.

6. Feel free to swing by my office, contact me by email, or call. I'd be happy to talk to you most anytime, but would appreciate setting up an appt. if you need a specific question addressed.

7. Course announcements will be posted on Canvas and some may be distributed by email. Check your email and the Canvas announcements regularly!

CSE Policies

1. The CSE Department has an anonymous contact form that you may use to voice your concerns about any problems in the course or department if you do not wish to be identified. (`http://cse.unl.edu/contact-form`)

2. CSE Department policy dictates that students in CSE courses are expected to regularly check their email so they do not miss important announcements.

3. Consider the Student Resource Center in Avery 12 if I am not available, or you otherwise need help: (`http://cse.unl.edu/src`)

4. All homework assignments, quizzes, exams, etc. must be your own work. No direct collaboration with fellow students, past or current, is allowed unless otherwise stated. The Computer Science & Engineering department has an Academic Integrity Policy (`http://cse.unl.edu/academic-integrity-policy`). All students enrolled in any computer science course are bound by this policy. You are expected to read, understand, and follow this policy. Violations will be dealt with on a case by case basis and may result in a failing assignment or a failing grade for the course itself.

UNL Policies

1. UNL's Academic Senate makes available a Syllabus Policy (`http://www.unl.edu/facultysenate/syllabuspolicy.pdf`), which lists elements that must be included in the syllabi of all UNL courses. This is also summarized by ASUN (`http://asun.unl.edu/content/syllabus-policy`).

2. Text on accommodations of students with disabilities:

   > Students with disabilities are encouraged to contact [the instructor or teaching assistant] for a confidential discussion of their individual needs for academic accommodation. It is the policy of the University of Nebraska-Lincoln to provide flexible and individualized accommodations to students with documented disabilities that may affect their ability to fully participate in course activities or to meet course requirements. To receive accommodation services, students must be registered with the Services for Students with Disabilities (SSD) office, 132 Canfield Administration, 472-3787 voice or TTY.

Schedule

This schedule is tentative and may change:

Week	Dates	Topic	Lab Assigned (on Thursdays)	Lab Due (on Thursdays)
1	Jan 9-13	Ringo Setup (no class Jan 10)	Lab 1	
2	Jan 16-20	Intro to RTS/Architectures	Lab 2	Lab 1
3	Jan 23-27	Interrupts	Lab 3	Lab 2
4	Jan 30-Feb 3	Shared Data		
5	Feb 6-10	Worst-Case Execution Time	Lab 4	Lab 3

6	Feb 13-17	Real-Time Scheduling		
7	Feb 20-24	Real-Time Scheduling	Lab 5	Lab 4
8	Feb 27-Mar 3	Real-Time Scheduling		
9	Mar 6-10	Shared Resources		
10	Mar 13-17	Shared Resources/Reentrancy	Lab 6	Lab 5
11	Mar 20-24	***Spring Break - No Class***		
12	Mar 27-31	RTOS Design Principles		
13	Apr 3-7	***Flexible Time***/Distributed RTS	HW 1	Lab 6
14	Apr 10-14	Distributed RTS		
15	Apr 17-21	***No Class (Dr. Bradley traveling)***		
16	Apr 24-28	Distributed RTS	Final Exam Assigned	HW 1

The final exam is Tuesday May 2, 2017 from 10:00 - 12:00.