Course Syllabus
COMP 737 – Real-Time Systems
Fall 2023

Meeting Place: SN 011
Meeting Time: 2:00 - 3:15, TuTh
Course Web Page: http://www.cs.unc.edu/∼anderson/teach/comp737. The powerpoint slides and other things can be found here.

Instructor: Prof. Jim Anderson
Telephone: 590-6057
Office: FB316
E-mail: anderson@cs.unc.edu
Office Hours: 12:30 - 1:30, TuTh

Homework Grader: Joseph Goh
E-mail: jgoh@cs.unc.edu

Goal of the Course: To study issues related to the design and analysis of systems with real-time constraints. The problem of ensuring such constraints is ultimately a scheduling problem, so much attention is devoted to such problems. At the end of the course, you will be able to prove analytical results pertaining to real-time scheduling algorithms and synchronization protocols that enable timing constraints to be validated. You will also be aware of implementation concerns that arise when deploying these algorithms and protocols in a real system. Finally, you will have an appreciation of the range of problems studied by researchers in the area of real-time systems. This is a "must" course for anyone wanting to do real-time systems research in this department.

Text: *Real-Time Systems*, Jane Liu, Prentice Hall, 2000.

We will discuss in class how to get access to this text. We will also use some papers from the literature. These papers and some additional "classic" papers can be found on the course web page.

Prerequisites: COMP 530 (Undergrad Operating Systems). In addition, one of the things researchers in this area need to be comfortable with is computational-complexity issues pertaining to validating timing constraints. Because of this, for a class or two, some basic knowledge of NP-completeness, as covered in COMP 750 and some undergrad classes, will be useful. That said, a deep understanding of NP-completeness is not necessary. *No NP-completeness-related questions will be asked on exams* (though some easy ones may come up on homeworks). **Please note that I do not intend to zealously enforce the prerequisites.** Anyone with a decent background in algorithms and operating systems should be able to handle the material.

Grading:	Homework	20%
	Project	30%
	Midterm Exam	20%
	Final Exam	30%

We will probably have five or six homework assignments. Some of these will involve programming. *All homework assignments must be completed individually*. These assignments are designed to make sure you're keeping pace: you should not find them extraordinarily time-consuming. *You must include a signed honor statement with each homework submission indicating that it is your own work.*

Each student must complete a class project. You are responsible for defining your own project. Your project can be either an experimental investigation or a survey or research paper. The project must be a fairly significant piece of work. While survey papers are perfectly fine, I'd encourage you to try to do something more research-oriented. It is perfectly fine to use research from an RA position as the basis

for your class project. However, your project may not be based on work from another course without the permission of me and the instructor for that course (permission will be granted only if the total work involved is commensurate with the amount of effort expected in both courses combined). Two-person projects may be permitted, provided the total work involved is about twice that of the typical single-person project. **Note:** In producing your project report, usage of ChatGPT (or similar AI programs) is not allowed and will be treated as an honor code violation.

The final exam will cover the entire course.

Note: I do re-use old homeworks and exam questions. Any attempts to access old homeworks and exams from the files of students who have taken this course previously, or from other sources, will be considered an honor code violation.

Class Etiquette: You are expected to maintain proper etiquette in class. This includes:

- not making a habit of arriving late, leaving in the midst of class, or skipping class,
- not talking, sleeping, reading newspapers, eating, etc. in class,
- keeping cellphones off,
- and not using your laptop to browse the web.

Note: Posting materials from this class online without prior authorization is an honor code violation.

Class Participation: This class will be far more enjoyable for everyone if all students come to class ready and willing to discuss the material to be covered. I plan to reward those who participate in class by increasing their final grade by up to half a letter grade. I also reserve the right to add a similar negative "reward" for those who do not observe appropriate etiquette in class.

Topics: The list of topics I plan to cover is given below. (Chapter numbers refer to Liu's book.) Note: We have 28 total classes, one of which will be used for the midterm.

Part I: Uniprocessor Scheduling of Independent Tasks.

- **Introduction to real-time systems (1 week).**
  - J. Stankovic, "Misconceptions About Real-Time Computing," *IEEE Computer*, Vol. 21, No. 10, October 1988, pp. 10-19.
  - Chapter 1: Example real-time applications.
  - Chapter 2: Hard vs. soft real time.
  - Chapter 3: Reference model (includes lots of definitions used in later chapters).
- **Classic uniprocessor scheduling results (4 weeks).**
  - Static scheduling.
    - Chapter 5: Cyclic executives.
  - Dynamic scheduling.
    - Dynamic-priority scheduling:
      - Chapter 4, Section 6: Optimality of EDF and LLF.
      - Chapter 6, Section 3: Utilization-based schedulability test for EDF.

· Nonpreemptive EDF from:
K. Jeffay, D. Stanat, and C. Martel, "On Non-Preemptive Scheduling of Periodic and Sporadic Tasks," *Proceedings of the 12th IEEE Real-Time Systems Symposium*, pp. 129-139, December 1991.

* Static-priority scheduling:
  · Chapter 6, Section 4: Optimality of RM and DM.
  · Chapter 6, Section 7: Utilization-based schedulability test for RM. (Skip 6.7.3 – 6.7.5.)
  · Chapter 6, Sections 5 and 6: Demand-based scheduling conditions for static-priority systems.
* Dealing with complexities arising in real systems.
  · Chapter 6, Section 8: Practical considerations. (Skip 6.8.6 – 6.8.7.)
  · Timing analysis, from:
  C.M. Krishna and K.G. Shin, *Real-Time Systems*, McGraw Hill, pp. 25-37.

- **Intractibility results (1.5 weeks).**
  - Preemptive systems.
    * Dynamic-priority systems, from:
    S. Baruah, R. Howell, and L. Rosier, "Feasibility Problems for Recurring Tasks on One Processor," *Theoretical Computer Science*, Vol. 118, pp. 3-20, 1993.

    F. Eisenbrand and T. Rothvoss, "EDF-schedulability of Synchronous Periodic Task Systems is coNP-hard," *Proceedings of the 21st ACM-SIAM Symposium on Discrete Algorithms*, pp. 1029-1034, January 2010.

    P. Ekberg and W. Yi, "Uniprocessor Feasibility of Sporadic Tasks with Constrained Deadlines is Strongly coNP-Complete," *Proceedings of the 27th Euromicro Conference on Real-Time Systems*, pp. 281-286, July, 2015.
      · The last two papers are a little complicated: their results will be stated without proof.
    * Static-priority systems, from:
    J. Leung and J. Whitehead, "On the Complexity of Fixed-Priority Scheduling of Periodic, Real-Time Tasks," *Performance Evaluation*, Vol. 2, No. 4, pp. 237-250, 1982.

    F. Eisenbrand and T. Rothvoss, "Static-Priority Real-Time Scheduling: Response Time Computation is NP-hard," *Proceedings of the 29th IEEE Real-Time Systems Symposium*, pp. 397-406, December 2008.
      · This paper is a little complicated: its results will be stated without proof.
  - Nonpreemptive systems.
    * Dynamic-priority systems, from:
    K. Jeffay, D. Stanat, and C. Martel, "On Non-Preemptive Scheduling of Periodic and Sporadic Tasks," *Proceedings of the 12th IEEE Real-Time Systems Symposium*, pp. 129-139, December 1991.
    * Static-priority systems (no good reference here).

Part II: Beyond Uniprocessor Independent Task Models.

- **Resource sharing (1 week).**
  - **Motivation:** "What Really Happened on Mars Rover Pathfinder," by Mike Jones.
  - Chapter 8: Priority inheritance and priority ceiling protocols, stack resource protocol. (Skip 8.7 – 8.10.)

  - Lock-free approach (very briefly), from:
    J. Anderson, S. Ramamurthy, and K. Jeffay, "Real-Time Computing with Lock-Free Objects," *ACM Transactions on Computer Systems*, Vol. 15, No. 6, pp. 388-395, May 1997.

- **Multiprocessor schedulability analysis (1 week)**.
  - General overview, mostly from:
    J. Carpenter, S. Funk, P. Holman, A. Srinivasan, J. Anderson, and S. Baruah, "A Categorization of Real-Time Multiprocessor Scheduling Problems and Algorithms," in *Handbook of Scheduling: Algorithms, Models, and Performance Analysis*, Joseph Y. Leung (ed.), Chapman and Hall/CRC, Boca Raton, Florida, pp. 30.1-30.19, 2004.
  - Hard real-time analysis for global EDF, from:
    S. Baruah, "Techniques for Multiprocessor Global Schedulability Analysis," *Proceedings of the 28th IEEE Real-Time Systems Symposium*, pp. 119-128, December 2007.
  - Soft real-time analysis for global EDF, from:
    U. Devi and J. Anderson, "Tardiness Bounds under Global EDF Scheduling on a Multiprocessor," *Real-Time Systems*, Vol. 38, No. 2, pp. 133-189, February 2008.

- **Synchronization in multiprocessors and distributed systems (1 week).**
  - Multiprocessor locking protocols, from:
    B. Brandenburg and J. Anderson, "Optimality Results for Multiprocessor Real-Time Locking," *Proceedings of the 31st IEEE Real-Time Systems Symposium*, pp. 49-60, December 2010.
  - Chapter 9, Section 4: End-to-end scheduling.

- **Mixing real-time and non-real-time (1 week).**
  - Chapter 7, Section 1: Introduction.
  - Chapter 7, Section 2: Deferrable servers.
  - Chapter 7, Section 3: Sporadic servers.
  - Chapter 7, Section 4: Constant utilization and total bandwidth servers. (We will skip weighted fair queuing, since we are covering proportional-share scheduling, which is similar.)

- **Fairness (1 week).**
  - Proportional-share scheduling, from:
    I. Stoica, H. Abdel-Wahab, K. Jeffay, S. Baruah, J. Gehrke, and C.G. Plaxton, "A Proportional Share Resource Allocation Algorithm for Real-Time, Time-Shared Systems," *Proceedings of the 17th IEEE Real-Time Systems Symposium*, pp. 288-299, December 1996.
  - Pfair scheduling, from:
    A. Srinivasan and J. Anderson, "Optimal Rate-based Scheduling on Multiprocessors," *Journal of Computer and System Sciences*, Volume 72, Issue 6, pp. 1094-1117, September 2006.

- **A quick look at some real systems (or, how much of this theory really gets used anyway?) (1 week).**
  - Chapter 12, Sections 1 and 2: Basic operating-system functions needed for real-time computing.
    * The above will be supplemented with some material from Chapter 2 of Björn Brandenburg's Ph.D. dissertation: *Scheduling and Locking in Multiprocessor Real-Time Operating Systems*, The University of North Carolina at Chapel Hill, 2011.
  - Chapter 12, Sections 6 and 7: A brief survey of commercial real-time and non-real-time operating systems.

* The above will also be supplemented with some material from Chapter 2 of Björn Brandenburg's Ph.D. dissertation.

Part III: Beyond This Class. We have a weekly "CPS (cyber-physical systems) lunch" meeting where ongoing research in CPS, including work by our real-time group, is discussed. You are welcome to attend these meetings. We also have weekly project meetings. You are welcome to attend those as well.

University Resources:

Accessibility Resources: The University of North Carolina at Chapel Hill facilitates the implementation of reasonable accommodations, including resources and services, for students with disabilities, chronic medical conditions, a temporary disability or pregnancy complications resulting in barriers to fully accessing University courses, programs and activities. Accommodations are determined through the Office of Accessibility Resources and Service (ARS) for individuals with documented qualifying disabilities in accordance with applicable state and federal laws. See the ARS Website for contact information: `https://ars.unc.edu` or email `ars@unc.edu`.

Counseling and Psychological Services (CAPS): CAPS is strongly committed to addressing the mental health needs of a diverse student body through timely access to consultation and connection to clinically appropriate services, whether for short or long-term needs. Go to their website: `https://caps.unc.edu/` or visit their facilities on the third floor of the Campus Health Services building for a walk-in evaluation to learn more.

Title IX Resources: Any student who is impacted by discrimination, harassment, interpersonal (relationship) violence, sexual violence, sexual exploitation, or stalking is encouraged to seek resources on campus or in the community. Please contact the Director of Title IX Compliance (Adrienne Allison Adrienne.allison@unc.edu), Report and Response Coordinators in the Equal Opportunity and Compliance Office (reportandresponse@unc.edu), Counseling and Psychological Services (confidential), or the Gender Violence Services Coordinators (gvsc@unc.edu; confidential) to discuss your specific needs. Additional resources are available at safe.unc.edu.