EECS 571 PRINCIPLES OF REAL-TIME COMPUTING Fall'10

Instructor: Kang G. Shin, 4605 CSE, 763-0391; kgshin@umich.edu

Number of credit hours: 4

Class meeting time and room: Regular classes: MW 10:30am – noon at EECS 1012
Makeup/Discussion classes: F 10:30am – 11:30am at EECS 1012). (will be announced a week in advance)

Office hours: MW 9:30 – 10:30am, or by appointment
Email is the best way to get hold of me.

Course homepage: http://eecs.umich.edu/courses/eecs571

Important Dates:

- Start of class: Sep. 8, Wednesday
- Fall study break: Oct. 18–19 (Mon–Tue)
- Midterm (comprehensive): Dec. 1, Wed (tentative).
- One-page project proposal due: Oct. 13, Wednesday
- Thanksgiving break: 5pm Nov. 24 (Wed)–8am Nov. 29 (Mon)
- Last day of class: Dec. 13 (Mon)
- Term project presentations: in class and evening of Dec. 13 (Mon)
- Study days: Dec. 14 (Tue) & Dec. 18–19 (Sat.–Sun.)
- Term project report due: Dec. 17, Friday

Newsgroup and email list: *umich.eecs.class.571* will be available for class discussions, questions and answers. You should subscribe to the mail list by sending email to *eecs571-request@eecs* and typing *subscribe* in the "Subject" field. The newsgroup or email list should not be used for announcing personal matters, or selling used cars or computers.

Course materials: "Real-Time Systems," Krishna and Shin, McGraw-Hill, 1997. (This book is out of stock and copies are available at Dollar Bill.)
The log of errata is maintained in
`http://www.eecs.umich.edu/courses/eecs571/book_correction.pdf`
and typos and other errors should be reported to `rtbook@tikva.ecs.umass.edu`.

Also used are additional articles from archival journals and conference proceedings. Three key sources are:

- IEEE Real-Time Systems Symposium (RTSS) (1980 –)
- IEEE Real-Time Technology and Applications Symposium (RTAS) (1995 –)
- International Journal of Time-Critical Computing (1989 –)

All of these proceedings are accessible electronically through the University Digital Library (http://www.ieeexplore.org).

Pre-requisites: EECS 482 or EECS 470, or basic knowledge in probability and random processes is required, or instructor's approval.

Grading policy:

Homework (approximately six assignments):	15%
Midterm (comprehensive) in class on Nov. 22, Monday (tentative):	25%
Term project (presentation 30% & report 25%):	55%
Class participation:	5%

Policy on Collaboration: You may, and are encouraged to, team up with another student (so each team of 1–2 members) to work on your term projects. Homeworks, on the other hand, must be done *individually*. However, I encourage you to actively participate in net discussions (using the newsgroup address) on clarifications and additional reading sources on lectures, HWs, and projects.

Acts of cheating and plagiarizing will be reported to the Engineering Honor Council. Cheating is when you copy, with or without modification, someone else's work that is not meant to be publicly accessible. Plagiarizing is when you copy, with or without modification, someone else's work that is publicly available without acknowledging the original source.

Regrading Policy: You have five (5) working days to ask for a regrade on everything. To ask for regrade, you must submit a written request explaining the technical reasons that would make a regrade necessary. Upon such a request your HW or project or exam may be regraded not only for the parts requested but also for the other parts.

Course Contents

Due to its vital role in almost all application domains, such as ground and air/space vehicles, robots, buildings and even human bodies, as well as telecommunication systems and devices, real-time computing (RTC) has become an essential discipline in the field of computer science and engineering. The new emerging concept of *cyber-physical system* (CPS) is also rooted at real-time computing.

This course is intended to cover principles and foundations (*not* case studies or applications) of real-time computing, which are based on three attributes: **timeliness**, **reliability/safety**, and **environmental interface**. These three attributes are strongly coupled with each other by a single precious resource, **time**, which is, in turn, dictated by limited resources, electric energy, space and weight. In this course, students will be exposed to the state-of-art (both analytical and experimental) research and development related to all these three attributes and their interplay.

Tentative Topics to Be Covered:

1. Background, motivation and definition of real-time computing.
2. Cyber-physical systems: concepts, examples, and issues
3. Characterization of real-time computing systems: Performance measures and deadlines.
4. Estimation the execution time of real-time tasks, and evaluation of the system's ability of meeting deadlines.
5. Task assignment and scheduling to meet deadlines.
6. Real-time OS and system software: requirements, principles, and examples.
7. Low-power embedded systems.
8. Time-sensitive communications: protocols and end-to-end delay guarantees, and their implementation.
9. Security and privacy of embedded systems and devices.
10. Model-based integration of embedded real-time software: algorithms, languages, tools and applications.
11. Formal methods for specification and verification of embedded real-time systems.
12. Fault-tolerance and evaluation techniques for RTC systems: models, algorithms and architectures for error detection, fault isolation, recovery.

Additional sources of reading:

IEEE Publications:

- IEEE/ACM Transactions on Networking
- IEEE Transactions on Computers
- IEEE Transactions on Software Engineering
- IEEE Transactions on Parallel and Distributed Systems
- IEEE Computer Magazine
- IEEE Software Magazine
- IEEE Communications Magazine
- IEEE Networks Magazine
- Proceedings of Annual IEEE INFOCOM
- Proceedings of Annual IEEE International Conference on Distributed Computing Systems
- Proceedings of Annual IEEE Fault-Tolerant Computing Symposium *Note:* The name of this conference is changed to Conference on Dependable Systems and Networks, effective as of 2000.
- ACM SenSys, IEEE SECON, MASS, IPSN; ACM CCS

ACM Publications:

- ACM Trans. on Computer Systems
- Proceedings of Annual ACM Embedded Software Systems (EMSOFT)
- Proceedings of Annual ACM SIGCOMM
- Proceedings of Annual ACM Sigmetrics Conf. on Measurement and Modeling of Computer Systems
- Proceedings of Bi-annual ACM Symposium on Operating Systems Principles
- Proceedings of Bi-annual ACM Symposium on Operating Systems Design and Implementation (OSDI)
- Proceedings of SenSys

Other Publications:

- Proceedings of the International Conference on Information Processing in Sensor Networks (IPSN)
- Proceedings of Annual USENIX Technical Conference
- Journal of Distributed and Parallel Computing (Academic Press).
- IBM J. of Research and Development
- IBM Systems Journal
- Journal of Guidance and Control.

Guidelines for Homework and Term Project

Homework

There will be bi-weekly homework assignments based on the specific topics to be covered during the corresponding two weeks. These assignments are paper reading and summaries. For the paper-reading assignments, each student is required to read and analyze the book chapter(s), or two (2) or more papers during each assignment period, and submit a computer-generated, less than 2-page long report including:

Cover page: Title of the topic covered, author's name, affiliation/address, phone number, and e-mail address, date of submission, and a brief summary.

Analysis and Critiques: The chapters and papers read must be critically analyzed here. Mere copying parts of the book chapter and papers is strongly discouraged.

If I were the author(s): State what you would do differently from the author(s) and why.

References: List the papers you read following the standard IEEE format, i.e., authors' names, title of the paper, name of the journal/conference proceedings, volume and number if applicable, page number, month and year of publication.

In addition to the bi-weekly assignments, you may be asked every now and then (at most once during the term) to present your assignment in class. (Those who do not present must actively participate in the discussion. This is where class participation counts.)

Term Project

This is a very important requirement for the course! Each group of no more than two students must complete a project of their choice related to the area of real-time embedded computing and *approved* by the instructor. Upon completion of a project, each group must write a final report following the format given below:

1. The first page must include the title of the project, author's name, affiliation/address, phone number, and e-mail address, date of submission, and an abstract summarizing the problem studied and the results obtained.
2. Introduction: an informal problem statement and its importance, critiques of related work done by others (citing appropriate references), justification of your approach, an outline of your approach, and introduction of the organization of your report.
3. Formal/detailed problem statement along with the assumptions used, and solution approaches taken.
4. Design and implementation details if it is a systemish project.
5. Analysis/evaluation (using models, simulation or experimentation) of the proposed solutions and comparison with others', if possible.

6. Conclusions and/or discussions: Brag about what you have done, confess the weakness of your solution(s)/approaches, state the problems encountered and list unresolved but related issues.

7. References: List the papers you read following the standard IEEE format, i.e., authors' names, title of the paper, name of the journal/conference proceedings, volume and number if applicable, page number, month and year of publication.

The students are encouraged to start thinking of the projects as early as possible and prepare project proposals, and have discussions with the instructor. The instructor will do his best in assisting the students for the selection and completion of each group's project. **A one-page proposal per group must be submitted to the instructor no later than Wednesday, October 13, 2010. Final project presentations are scheduled during 6pm-midnight, December 13, 2010. Typed final project reports must be submitted *electronically* at or before noon, Friday, December 17, 2010.** You are strongly encouraged to arrange meetings with the instructor to discuss your project topics and proposals before and after the proposal due date (and of course, throught the semester).

Note that selection of project partners is totally up to individual groups. Since the members of a group will usually receive the same project grade, it is important for each of you to choose good partners. One-person groups are perfectly OK, but cannot be used as an excuse to reduce the scope of the project.

In addition to submitting a written report, each group may be required to make a 20-minute demo or oral presentation in the evening of December 13, 2010. Especially, the correctness of implementation-oriented projects must be demonstrated in front of the instructor, or the evidences of their correctness must be documented in the final report.

Given below is a list of potential project topics (this list is expected to be updated during the semester):

1. Development of mathematical models and their solutions for the performance and reliability of real-time systems, and evaluation of the solutions' effectiveness using "unbiased" simulations or implementation.

2. Development of new resource management policies (as part of OS or middleware) for real-time systems and their evaluation; or implementation/simulation of several competing policies and their in-depth comparison.

3. Development of tools for simulation or debugging tool for real-time systems, and collection and analysis of experimental data from simulators or real systems. High performance and high reliability architectures.

4. Detailed evaluation of real-time network protocols and proposals to remedy their deficiencies in dealing with timeliness issues.

5. Design and evaluation of real-time networking (especially Internet and embedded) applications like video conferencing, automated factory communications, etc.

6. Architectural support of OS time management, time-constrained intertask communications, I/O-intensive OS functions (e.g., high-speed interrupt handling), system monitoring.

7. Algorithms and architectures for scheduling tasks and intertask messages to meet deadlines.

8. Real-time OS and programming languages: how to express and meet timing requirements while retaining the salient features of the non-real-time counterparts, when possible.

9. Algorithms, architectures, and OS/middleware implementations for error detection, fault location, and recovery subject to deadline constraints.

10. Theory and tools for formal specification and verification of real-time systems.

11. Sensor networks and their applications.

12. Case studies: Video-on-demand systems, telecommunications, embedded real-time control (e.g., robot/aircraft control, control and monitoring of computer-integrated manufacturing systems, drive-by-wire), animation and real-time graphics, real-time artificial intelligence, on-line transaction systems, intelligent transportation systems.