Carleton University
Department of Systems and Computer Engineering

SYSC 3303 A **Real-Time Concurrent Systems** **Fall 2016**

Course Outline

Instructor: Dr. Lynn Marshall, Room ME4230, lynnmar@sce.carleton.ca

Course Objectives

To introduce students to the principles and practice of software development for systems characterized by one or more of the following terms: real-time, concurrent, event-driven, and embedded. Although a specific implementation technology will be used to provide hands-on programming experience, the goal is to present techniques that are applicable to a diverse range of applications, hardware/software components, programming languages and operating systems.

Learning Outcomes

By the end of this course students should be able to:

1) Understand the ISO Protocol Stack with emphasis on UDP inter-process communication.
2) Model the structure and behaviour of a concurrent system using Use Case Maps and UML.
3) Write multi-threaded communicating programs in Java.
4) Understand and apply the theory of cyclic executives, rate-monotonic analysis, and priority-based scheduling of a real-time concurrent system.
5) Design, implement, test, and document a reasonably complex and large concurrent system using a development process based on incremental milestones.
6) Work in a team using industrial engineering tools, including version control, development, testing and debugging environments.

Prerequisite Courses

For Engineering students, the prerequisites are: SYSC 2004 or SYSC 2100, **and** SYSC 2003.

Computer Science students must have successfully completed COMP 2003, **and** COMP 2002 or COMP 2402.

Information Technology, University of Ottawa Engineering students, and Carleton University Special students must meet the requirements of the Systems and Computer Engineering department.

Prerequisite waivers will not be granted. Students who have not completed the prerequisites must withdraw from SYSC 3303 by the last date for registration in early summer term courses; otherwise, they will be deregistered before the end of term.

Textbook

There is no required textbook. As the term progresses, lecture slides will be posted as PDF files on the course Web site. Note that additional material that is **not** on the posted slides will be presented in class.

Optional textbook: *Real-Time Systems and Programming Languages: Ada, Real-Time Java and C/Real-Time POSIX*, Fourth Edition, Alan Burns and Andy Wellings, Addison-Wesley, 2009.

Additional supplementary references will be listed on the course web site.

Web Site

Course materials will be placed on the SYSC 3303 Web site, so students must have Internet access. The URL for the site is http://www.sce.carleton.ca/courses/sysc-3303/f16. Students are expected to access this site regularly, and consult it before emailing questions to the instructor. Portions of the Web site will be protected by a password, which will be provided to students in class. Please do not make this password publicly available.

Attendance

Students are expected to attend all lectures and labs. The Faculty of Engineering and Design requires students to have a conflict-free timetable, so requests to accommodate missed exams, assignment due dates, project milestones, etc., because of conflicts with other courses, jobs, or vacation plans will not be considered.

Assignments

There will be five assignments. Late assignments will not normally be accepted; however, students who cannot submit an assignment by the due date for valid medical or compassionate reasons should contact the instructor immediately to arrange appropriate accommodations (e.g. an extension of the due date).

Assignment #3 will involve summarizing an article or articles on current research or work in real-time concurrent systems. Students who elect to give a 10-15min in-class presentation on their summary will earn up to 3% in bonus marks.

Labs

Attendance at lab periods will be mandatory. Teams will have meetings with a TA in many of the labs.

Project

A major component of the course is a team project, done in groups of 4 or 5, which will lead you through the process of building a reasonably complex concurrent system. Each team member must participate in all aspects of the project: design, coding, testing and debugging, etc. You will get your team's mark for the project, multiplied by a factor of between 0% and 150%, based on the instructor's and the TAs' judgement of your contribution to your team. Students who do not pull their weight on their project team and projects submitted by students who refuse to join a team will receive a mark of 0; these students will receive a final grade of FND regardless of how well they do in the other course components.

Exams

One closed-book midterm exam will be held during the lecture on Wed Oct 12th.

Students who are unable to write the midterm exam because of illness or other circumstances beyond their control must provide, in cases of illness, a medical certificate dated no later than one working day after the exam, or appropriate documents in other cases. If this information is provided to the instructor no later than five working days after the exam, the student will write an essay that will be used as the exam mark; otherwise, the mark for the missed exam will be 0.

A closed-book final exam will be held during the University's December examination period. With the exception of those students who receive an FND based on their participation in the team project (see the "Projects" section), **and** those who did not write the midterm exam OR essay, all students are eligible to write the final examination. Eligible students, who miss the final exam and received at least 40% on the midterm exam, will receive the grade ABS and may apply to the Registrar's Office for deferral of the final examination. Eligible students who miss the final exam and received less than 40% on the midterm will receive a grade of FND and be ineligible for the deferred final exam.

Evaluation and Grading Scheme

Students will be evaluated by means of assignments, a project, a midterm exam, an optional in-class presentation, and a final exam. At least two of the questions on the final exam will be clearly labeled "Core Programming". These questions will be used to evaluate the students' understanding of the fundamental programming skills taught in this course. To pass the exam, students must (1) achieve an average mark on the "Core Programming" questions of at least 50%, and (2) achieve an overall final exam grade of at least 50%. If these requirements are met, the student's grade is calculated as follows:

Assignments: 10% (2% each)
Midterm Exam: 15%
Project: 35%
Final Exam: 40%
Optional In-Class Presentation of Assignment #3: up to 3% bonus marks
Optional Centre for Student Academic Support Workshop Attendance: up to 2% bonus marks (one per workshop)

Students with Disabilities

The Paul Menton Centre for Students with Disabilities (PMC) provides services to students with Learning Disabilities (LD), psychiatric/mental health disabilities, Attention Deficit Hyperactivity Disorder (ADHD), Autism Spectrum Disorders (ASD), chronic medical conditions, and impairments in mobility, hearing, and vision. If you have a disability requiring academic accommodations in this course, please contact PMC at 613-520-6608 or pmc@carleton.ca for a formal evaluation. If you are already registered with the PMC, contact your PMC coordinator to send me your ***Letter of Accommodation*** at the beginning of the term, and no later than two weeks before the first in-class scheduled test or exam requiring accommodation (*if applicable*). **Requests made within two weeks will be reviewed on a case-by-case basis.** After requesting accommodation from PMC, meet with me to ensure accommodation arrangements are made. Please consult the PMC website (www.carleton.ca/pmc) for the deadline to request accommodations for the formally-scheduled exam (*if applicable*).

Health and Safety

Every student should have a copy of our Health and Safety Manual. An electronic version of the manual can be found at http://www.sce.carleton.ca/courses/health-and-safety.pdf.

Week by Week Outline: Topics Covered

- Nature of Real-Time Systems
- Internet Protocols
- The TFTP Protocol, Part 1
- Java Threads
- Synchronization of Java Threads
- The TFTP Protocol, Part 2
- UML
- Lifecycle and Scheduling of Java Threads
- Use Case Maps (UCMs)
- Verification and Validation
- Scheduling Schemes for Real-Time Systems
- Recent Developments in Real-Time Concurrent Systems