Carleton UNIVERSITY | Department of Systems and Computer Engineering

SYSC 3303
Real-Time Concurrent Systems

Calendar description

Principles and practice of a systems engineering approach to the development of software for real-time, concurrent, distributed systems. Designing to achieve concurrency, performance, and robustness, using visual notations. Converting designs into programs. Introduction to hard real-time systems. Team project.

Includes: Experiential Learning Activity.

Lectures three hours a week, laboratory two hours a week

http://calendar.carleton.ca/undergrad/courses/SYSC/

Prerequisites

For students in the Faculty of Engineering and Design: (SYSC 2003 or SYSC 3310) and SYSC 2004. For students in Computer Science: COMP 2401 and COMP 2402.

Prior knowledge

This course involves a substantial amount of programming, both for assignments and a team project. Students are expected to know how to write and debug sequential programs using the programming language Java.

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.

List of topics

- Nature of Real-Time Systems
- Concurrency
- Java Threads
- Synchronization of Java Threads
- Internet Protocols
- UML

- Real-Time Software Design Method
- Analysis of Real-Time Software Designs
- Verification and Validation
- Recent Developments in Real-Time Concurrent Systems

Learning outcomes

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

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

Graduate Attributes (GAs)

The Canadian Engineering Accreditation Board requires graduates of engineering programs to possess 12 attributes at the time of graduation. Activities related to the learning outcomes listed above are measured throughout the course and are part of the department's continual improvement process. Graduate attribute measurements will not be taken into consideration in determining a student's grade in the course. For more information, please visit: https://engineerscanada.ca/.

Graduate Attribute	Learning outcome(s)
1.4.S: Knowledge Base: Applied: Programming and algorithms	1, 2, 3, 5
2.2: Problem Analysis: Developed: Approach to the problem	6
3.3: Investigation: Developed: Developed: Experimental procedure	4
3.4: Investigation: Developed: Data reduction methods and results	4
3.5: Investigation: Developed: Interpretation of data (synthesis) and discussion	4
4.4: Design: Developed: Design solution(s)	2, 6
4.5: Design: Developed: Design implementation / task(s) definition	2, 3, 5, 6
6.1: Individual and Team Work: Developed: Personal and group time management	7
6.2: Individual and Team Work: Developed: Group culture, group dynamics	7
6.3: Individual and Team Work: Developed: Leadership: initiative and mentoring, areas of expertise, and interdisciplinary teams	7

Accreditation Units (AUs)

For more information about Accreditation Units, please visit: https://engineerscanada.ca/.

The course has a total of 49 AUs, divided into:

- Engineering Science: 60%
- Engineering Design: 40%

Instructor and TA contact

Specific to course offering (tbd)

Textbook (or other resources)

Specific to course offering (tbd)

Evaluation and grading scheme

Specific to course offering (tbd)

Breakdown of course requirements

Specific to course offering (tbd)

Tentative week-by-week breakdown

Specific to course offering (tbd)

General regulations

Specific to course offering (tbd)