Syllabus: ECE652 Safety-critical Real-time Software

Sebastian Fischmeister
Electrical and Computer Engineering
University of Waterloo
sfischme@uwaterloo.ca

March 8, 2019

Course Title: Safety-critical Real-time Software

Course Short Title: Safety-crit RT Software

Instructor: Prof. Sebastian Fischmeister

1 Calendar Description

Concepts, theory, tools, and practice to understand, design, and write embedded software. This covers topics such as embedded computing hardware, embedded computing software, modeling of timing and real-time systems, dependability and safety, security, validation, and performance of embedded systems.

2 Prerequisite Topics

computer architecture, operating systems, programming languages, C

3 Prerequisite Courses

- Operating systems course
- Compilers course
- Programming course

4 Antirequisite Courses

- ECE455 Embedded Software

5 Lab Description

No lab this year

6 Tutorial Description:

No tutorials this year

7 Project

None this year

8 Reading Material

Disclaimer: Not all material listed. See the headings for the slide material.

- Wolf, Marilyn. Computers as Components: Principles of Embedded Computing System Design. Elsevier, 2012. Chapter 1-4
- Jane Liu. Real-time Systems, Kluwer, 2000. Chapter 1-6
- Lee, Edward Ashford, and Sanjit A. Seshia. Introduction to Embedded Systems: A Cyber-physical Systems Approach. MIT Press, 2016. Chapter 12-15
- Lyu, Michael R. Software Fault Tolerance. John Wiley & Sons, Inc., 1995. Chapter 1-5
- Seacord, Robert C. Secure Coding in C and C++. Pearson Education, 2005. Chapters 1-9 (exclude C++ portions)
- Shostack, Adam. Threat Modeling: Designing for Security. John Wiley & Sons, 2014. Chapter 2-3
- Jain, Raj. The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling. John Wiley & Sons, 1990. Chapter 1-11

- Thorsten Groetker, Ulrich Holtmann, Holger Keding, Markus Wloka. The Developer's Guide to Debugging. Springer. 2008. Chapter 2
- Laprie, Jean-Claude. Dependability: Basic concepts and terminology. Dependability: Basic Concepts and Terminology. Springer, Vienna, 1992. 3-245.
- Lewis, Daniel Wesley. Fundamentals of Embedded Software: Where C and Assembly Meet with Cdrom. Prentice Hall PTR, 2002. Chapter 6
- J. Regehr. Safe and Structured Use of Interrupts in Real-Time and Embedded Software. In Handbook of Real-time and Embedded Systems. 2008
- Avizienis, Algirdas, Jean-Claude Laprie, and Brian Randell. "Dependability and its threats: a taxonomy." Building the Information Society. Springer, Boston, MA, 2004. 91-120.
- Lamport, Leslie, Robert Shostak, and Marshall Pease. "The Byzantine Generals Problem." ACM Transactions on Programming Languages and Systems (TOPLAS) 4.3 (1982): 382-401.
- Fischmeister, Sebastian, and Insup Lee. "Temporal Control in Real-time Systems: Languages and Systems." Handbook of Real-Time Systems and Embedded Systems by (2007).
- Arafa, P., D. Solomon, S. Navabpour, and S. Fischmeister, "Debugging Behaviour of Embedded-Software Developers: An Exploratory Study", IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), Raleigh, USA, 2017.

Examples taken from:

- Laplante, Phillip A., and Seppo J. Ovaska. Real-time systems Design and Analysis: Tools for the Practitioner. John Wiley and Sons, 2011.

Extra reading:

- Driscoll, Kevin, et al. "Byzantine Fault Tolerance, from Theory to Reality." International Conference on Computer Safety, Reliability, and Security. Springer, Berlin, Heidelberg, 2003.
- Rushby, John. "Critical system properties: Survey and Taxonomy." Reliability Engineering & System Safety 43.2 (1994): 189-219.

9 Major topics

- **Part 1: Embedded computing**
  - Lecture syllabus
  - Embedded systems are difficult
  - Instruction sets
  - Computing units
  - Embedded computing platforms
  - Embdded system debugging
- **Part 2: Timing**
  - Reference model for real-time system
  - Common approaches to scheduling
  - Worst-case execution time problem
  - Clock-driven scheduling
  - Priority-driven scheduling
  - Programming timing constraints
- **Part 3: Safety**
  - Overview of dependable systems
  - Software fault tolerance
  - Interrupt scheduling
  - Byzantine Generals
  - Basics of software qualification
  - Basics of safety assessment
  - ISO 26262 & IEC 62304
- **Part 4: Security**
  - Security overview
  - STRIDE/DREAD
  - String vulnerabilities
  - Integer vulnerabilities
  - Formatted output vulnerabilities
  - Concurrency vulnerabilities
  - Recommended security practices
- **Part 5: Correctness analysis**
  - Invariants and temporal logic
  - Equivalence and refinement
  - Reachability and model checking
- **Part 6: Performance analysis**
  - Introduction to benchmarking
  - Common mistakes
  - Performance metrics
  - Probes and monitors
  - Ratio games and visualization + demo

10 Grading

Item	Weight
Quiz 1-3	50
Final	50

- Best two of the three quizzes will count.
- You have to achieve at least 50% in each category
- In borderline cases, based on the instructor's assessment, points can be traded 3:1.

11 Disclaimer

1. Academic Integrity

   In order to maintain a culture of academic integrity, members of the University of Waterloo community are expected to promote honesty, trust, fairness, respect and responsibility. Check www.uwaterloo.ca/academicintegrity/ for more information.

2. Grievance A student who believes that a decision affecting some aspect of his/her university life has been unfair or unreasonable may have grounds for initiating a grievance. Read Policy 70, Student Petitions and Grievances, Section 4, www.adm.uwaterloo.ca/infosec/Policies/policy70.htm. When in doubt please be certain to contact the department's administrative assistant who will provide further assistance.

3. Discipline

   A student is expected to know what constitutes academic integrity [check www.uwaterloo.ca/academicintegrity/] to avoid committing an academic offence, and to take responsibility for his/her actions. A student who is unsure whether an action constitutes an offence, or who needs help in learning how to avoid offences (e.g., plagiarism, cheating) or about 'rules' for group work/collaboration should seek guidance from the course instructor, academic advisor, or the undergraduate Associate Dean. For information on categories of offences and types of penalties, students should refer to Policy 71, Student Discipline, www.adm.uwaterloo.ca/infosec/Policies/policy71.htm. For typical penalties check Guidelines for the Assessment of Penalties, www.adm.uwaterloo.ca/infosec/guidelines/penaltyguidelines.htm.

4. Appeals

   A decision made or penalty imposed under Policy 70 (Student Petitions and Grievances) (other than a petition) or Policy 71 (Student Discipline) may be appealed if there is a ground. A student who believes he/she has a ground for an appeal should refer to Policy 72 (Student Appeals) www.adm.uwaterloo.ca/infosec/Policies/policy72.htm.

5. Note for Students with Disabilities

   Please register with the office and they will send me a notice.

12 Notes

- June 14, 2018: updated the reading material list to point to the book chapter of J. Regehr instead of the paper.