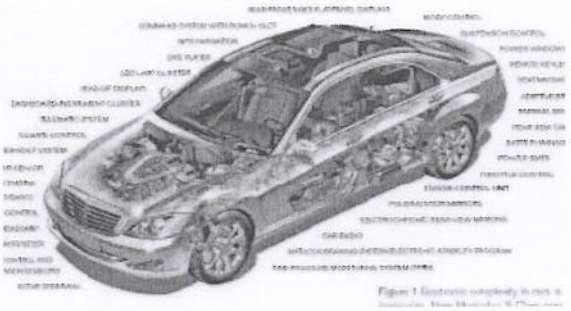
Virginia Tech

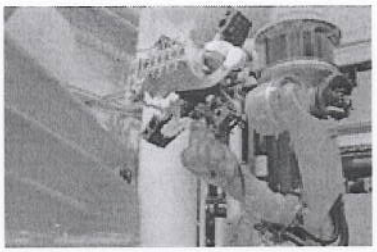
Virginia Tech ❖ Bradley Department of Electrical and Computer Engineering

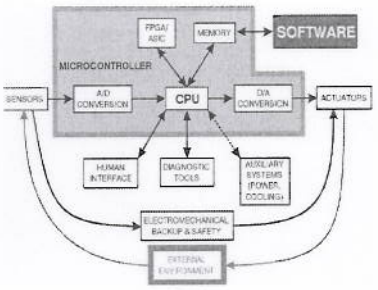
ECE 5550G Advanced Real-Time Systems

Course Syllabus



Instructor:
Dr. Binoy Ravindran
Associate Professor
ECE Dept., Virginia Tech
Office: 2040-C Torgersen Hall
Blacksburg, VA 24061
540-231-3777, binoy@vt.edu
http://www.ece.vt.edu/faculty/ravindran.html
http://www.real-time.ece.vt.edu/





Course Objectives:

Real-time computing has become an important sub-discipline of Computer Engineering. Many applications such as industrial automation (process control and discrete manufacturing), automotive, avionics, robotics, biomedical devices, and telecommunications require real-time computing. The main objectives of the course are to develop a strong understanding of the broad concept of real-time systems, obtain a practical understanding for the industry, and to stimulate research interest. The course presents the underlying theory, concepts, and practice of real-time systems. Main topics of the course include an introduction to real-time systems, real-time scheduling and resource management, real-time operating systems and kernels, real-time programming languages, and real-time application development using experimental or commercial real-time operating systems/middleware. The course enables students to design, analyze, and implement real-time systems, and prepares them for careers in real-time computing and related fields (e.g., embedded systems and safety-critical systems).

Upon completion of the course, the student should be able to:

1. Demonstrate real-time scheduling and resource management algorithms and protocols that enable timeliness requirements of real-time systems to be satisfied
2. Conduct real-time schedulability analysis
3. Design and implement real-time scheduling and resource management algorithms and protocols in real-time operating system kernels/real-time middleware, and measure timeliness performance
4. Demonstrate facilities available in real-time programming languages/operating systems for programming real-time systems, including the real-time POSIX standard and the Real-Time Specification for Java (RTSJ)
5. Design and implement real-time and embedded system applications using experimental/commercial real-time operating systems/middleware

Prerequisites:
Graduate Standing. Students are assumed to have taken an operating systems course during their undergraduate curriculum (students without an OS course should discuss with instructor). Strong knowledge of C, C++, or Java is also required.

Required Text:
Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications, Giorgio C. Buttazzo, Springer-Verlag, Second Edition, 2005, ISBN: 0-387-23137-4

Papers from the literature will be made available on the course website.

Operating System/Middleware Platform:

Several programming platform choices (for the course projects) are possible:

(a) The QNX Neutrino real-time operating system augmented with a thread scheduling middleware software developed at Virginia Tech, called "meta-scheduler." The professional edition of Neutrino with the QNX Momentics development suite is available to students of the course (free of charge, under an educational license with QNX). This edition of Neutrino and the meta-scheduler will be distributed to course students during the semester. Details of Neutrino can be found at: http://www.qnx.com/products/neutrino_rtos/.

(b) The real-time Linux kernel – i.e., the Linux kernel (http://www.kernel.org/) patched with real-time features (http://rt.wiki.kernel.org/index.php/Main_Page). This real-time Linux kernel is augmented with a kernel-level thread scheduling and resource management framework developed at Virginia Tech. This patched kernel is available to course students and will be distributed during the semester.

(c) Reference Implementations (RIs) of Sun Microsystem's Real-Time Specification for Java (RTSJ). Several RIs are freely available, including Sun's Real-Time Java Virtual Machine: http://java.sun.com/javase/technologies/realtime/rts/

Grading (tentative):

Programming Projects	60-64%
Homework	16-12% (3-4 homeworks)
Midterm Exam	12%
Final Exam	12%

Notes:

1. Three to four programming projects (with different grade percentages) are planned. The projects will involve implementation of real-time scheduling algorithms and resource access protocols.
2. It will be possible to substitute the final exam with a project that will involve substantial implementation and suggested by students (possible topics will be discussed during the course).

Course Topic Outline (tentative):

Topic	Buttazzo
Introduction to Real-time Systems	Chapter 1
Basic Concepts	Chapter 2
Aperiodic Task Scheduling	Chapter 3
Periodic Task Scheduling	Chapter 4
Fixed Priority Servers	Chapter 5
Handling Overload Conditions	Chapter 8, Lecture notes
Resource Access Protocols	Chapter 7
Real-time Operating Systems (Kernel Design, Example RTOSes)	Chapters 9, 11, Lecture notes
Real-Time POSIX	Lecture notes
Real-time Programming Languages (Real-Time Java)	Lecture notes