Master Course Description for EE-472

Title: Real-Time and Embedded Operating Systems

Credits: 4

EE 472: Real-Time and Embedded Operating Systems A software-intensive course in modern operating systems, with a focus on real-time and embedded applications. Lectures and programming assignments cover a range of topics from the classical OS concepts to real-time operating systems, including the OS kernel--process and task abstraction, scheduling, concurrency, memory management, file systems and IOs, RTOS (real-time operating systems), and case studies of RTOS programming for Bluetooth or IoT networking. **Prerequisite**: CSE 373 and CSE 374.

Coordinator: C. J. Richard Shi, Professor, Electrical and Computer Engineering

Goals: To introduce basic operating system concepts and development, acting as a major system-software course for Electrical and Computer Engineering students. To learn the basic components of real-time operating systems, including process and task abstraction, system kernel and APIs, synchronization and scheduling, storages and file systems. To prepare students with concepts and skills for system software development especially for embedded real-time applications.

Learning Objectives: At the end of this course, students will be able to:

1. Learn fundamentals of operating systems
2. Learn how to build your own real-time operating system
3. Learn how to design, develop and debug C code
4. Learn about modular design

Textbook:

Embedded and Real-Time Operating Systems, K.C. Wang, 2017 (1st Edition), Springer.

References:

Operating Systems: Principles and Practice (2nd Edition), by Tom Anderson and Mike Dahlin.

Prerequisites by Topic:

1. Mature knowledge of computer programming, and the C language (CSE 374).
2. Data structures and algorithms (CSE 373).

Topics:

1. Introduction to embedded systems & the ARM architecture
2. Exceptions, interrupts, device drivers
3. Embedded system models, event-driven embedded systems, process and formal models of embedded systems.
4. Process management and process synchronization.
5. Scheduling.
6. Memory management, memory protection, virtual memory.
7. Kernel mode and user mode processes, system calls.
8. Process management, memory management, device drivers, file systems.
9. Real-time embedded systems & OS

Course Structure and Programming Assignments: The class meets for three 50-minute lectures and one 50-minute TA-led lab session for students to work on programming assignments. For OS, students will learn as much from programming assignments as from lectures. During the quarter, students will work, in teams of two, on several programming assignments. Each lab will add a few features to a baseline primitive OS. By the end of the quarter, students will have built their own "complete" OS that can boot on real hardware, and is capable of scheduling multiple tasks, supporting a relatively full-featured virtual memory system, adding transactional updates to the file system, and building a bluetooth network.

Computer Resources: Students will use their own laptops for code development, or PCs in the department computer labs.

Grading: Approximate distribution: Programming Assignments 50%, Midterm 15%, Final Exam 30%, and 5% class participation.

ABET Student Outcome Coverage: This course addresses the following **ABET** outcomes:

H = high relevance, M = medium relevance, L = low relevance to course.

1. An ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics (H).
2. An ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors (M)
5. An ability to function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks, and meet objectives (M).
6. An ability to develop and conduct appropriate experimentation, analyze and interpret data, and use engineering judgment to draw conclusions (H).
7. An ability to acquire and apply new knowledge as needed, using appropriate learning strategies (H).

Prepared By: C.J. Richard Shi, Scott Hauck

Last Revised: May 13, 2020