E&C-Eng 570/670: (Advanced) System Software Design

Course Syllabus

General Info

Instructor:	David Irwin
Email:	irwin@ecs.umass.edu
Office:	Knowles Engineering Building 211D
Office Hours:	Thursdays 10am - 11am, or by appointment
Class Meetings:	WF 2:30pm - 3:45pm
Webpage:	http://www.ecs.umass.edu/ece570
Suggested (not required) Textbook:	OSTEP: http://pages.cs.wisc.edu/~remzi/OSTEP/

This syllabus covers both ECE570 and ECE670, which are taught in parallel. ECE670 differs from ECE570 in that it requires students to complete an independent semester-long research project, as described below, and it requires students to read and submit reviews of selected research papers.

Prerequisites

The prerequisites for ECE570 are ECE 331 (Hardware Organization and Design), formerly ECE232, and ECE 241 (Advanced Programming I), formerly ECE242. CS 187 (Programming with Data Structures) may be taken in lieu of ECE 242.

Course Description

This course covers systems software design and implementation at the graduate level. Software systems mediate the interaction between hardware and software by i) securely *multiplexing* hardware resources between multiple independent tasks and ii) providing *abstractions* that simplify software development. Topics include: process management, threading, synchronization, deadlock, scheduling, security, file systems, and device I/O. The course will also include additional research readings on embedded systems concepts, hardware virtualization, multicore support, and security. The concepts in this course are not limited to any particular software system or hardware platform. We will discuss examples that are drawn from many historically significant and modern software systems.

Objectives

The objective of this course is threefold: to demystify the seemingly complex interactions between software and hardware, to familiarize students with advanced issues in the design and

implementation of modern software systems, and to explain how systems design principles apply to the design of computing systems in general.

Grading Policy

Grades for 570 will be computed as follows:

50%	Assignments and Programming Projects
50%	Mid-term (25%) and Final Exam (25%)

Grades for 670 will be computed as follows:

45%	Assignments and Programming Projects
20%	Independent Research Project
35%	Mid-term (15%) and Final Exam (20%)

Class Participation

Lectures are mandatory and you are expected to attend regularly. One goal of this course is to promote the discussion of the issues in systems design among all class members. As such, you are encouraged and expected to ask questions, point out weaknesses in papers, and make observations during class.

Assignments and Programming Projects

There will be approximately 4 programming projects covering scheduling, threading, process organization, and security. Details will be available on the course webpage. The first project is simple, and is only meant to help you gain some experience with C++. The other projects will require a substantial time commitment on your part. However, you will be given approximately 2-3 weeks to complete each one. It is essential that you start the assignments early and not wait until the last minute! To complete the projects in this course, you will need the ability to develop software programs using C/C++. If you have not used C/C++ recently, you may want to refresh your knowledge using one of the many good books on the topic. In particular I recommend the classic, The C Programming Language, by Kernighan and Ritchie, because it is short and simple. There are also many online tutorials that Google can help you find. If you feel that you need extra help, please come see me. If you are not familiar with the UNIX computing environment, talk to me as soon as possible so we can bring you up to speed on what you need to know.

Students enrolled in 670 will also be required to write paper reviews for selected research papers throughout the course of the semester.

Independent Project Outline (670 only)

Students will select and work on a semester-long research project in consultation with the Professor. At the conclusion of the semester, students will submit a conference-style paper (10-12 pages) describing the results of their project. The project will include both an implementation component, which requires designing a software system, and an analysis component, which requires measuring the software system's performance under a chosen workload.

Midterm and Final Exam

There will be a written midterm examination in this course. The exam will be closed book, closed notes, and will stress a conceptual understanding of the material. Details regarding the specific format of the exam will be discussed in class. The final exam will be comprehensive covering material from the entire course with an emphasis on untested topics.

Lecture Topics

See calendar on website.

Accommodation Policy Statement

The University of Massachusetts Amherst is committed to providing an equal educational opportunity for all students. If you have a documented physical, psychological, or learning disability on file with Disability Services (DS), Learning Disabilities Support Services (LDSS), or Psychological Disabilities Services (PDS), you may be eligible for reasonable academic accommodations to help you succeed in this course. If you have a documented disability that requires an accommodation, please notify me within the first two weeks of the semester so that we may make appropriate arrangements.

Academic Honesty Policy

It is expected that all students will abide by the Academic Honesty Policy (`http://www.umass.edu/dean_students/academic_policy`), summarized below.

All members of the University community must participate in the development of a climate conducive to academic honesty. While the faculty, because of their unique role in the educational process, have the responsibility for defining, encouraging, fostering, and upholding the ethic of academic honesty, students have the responsibility of conforming in all respects to that ethic.

Intellectual honesty requires that students demonstrate their own learning during examinations and other academic exercises, and that other sources of information or knowledge be appropriately credited. Scholarship depends upon the reliability of information and reference in the work of others. Student work at the University may be analyzed for originality of con-

tent. Such analysis may be done electronically or by other means. Student work may also be included in a database for the purpose of checking for possible plagiarized content in future student submissions. No form of cheating, plagiarism, fabrication, or facilitating of dishonesty will be condoned in the University community. Academic dishonesty includes but is not limited to:

- **Cheating.** Intentional use or attempted use of trickery, artifice, deception, breach of confidence, fraud and/or misrepresentation of one's academic work
- **Fabrication.** Intentional and unauthorized falsification and/or invention of any information or citation in any academic exercise
- **Plagiarism.** Knowingly representing the words or ideas of another as one's own work in any academic exercise. This includes submitting without citation, in whole or in part, prewritten term papers of another or the research of another, including but not limited to commercial vendors who sell or distribute such materials
- **Facilitating Dishonesty** Knowingly helping or attempting to help another commit an act of academic dishonesty, including substituting for another in an examination, or allowing others to represent as their own one's papers, reports, or academic works

Sanctions may be imposed on any student who has committed an act of academic dishonesty. Any person who has reason to believe that a student has committed academic dishonesty should bring such information to the attention of the appropriate course instructor as soon as possible.