CSE 321 Realtime and Embedded Operating Systems Fall 2007

Course Description

Computers are embedded in countless real-world devices. These devices are required to perform flawlessly and in real-time. This course will address some of the fundamental challenges in the design, implementation, and validation of these real-time and embedded systems. Course topics include resource management, concurrency, secure coding practices, memory management, timeline design and analysis using metrics and schedulability tests, hardware interfacing, device driver programming, memory maps and boot kernels, firmware and ROM-resident system code, communications and networking, and debugging live systems. These concepts will be reinforced through C programming assignments using the Linux operating system.

On completion of this course students will be able to (i) understand the components and working of a realtime and embedded operating systems and (ii) design and implement various embedded operating system functions.

Course Information

Website:	http://www.cse.buffalo.edu/~bina/cse321/fall2007
Instructor:	Bina Ramamurthy (bina@cse.buffalo.edu)
Lecture Time:	TTh: 11.00AM – 12.20PM
Lecture Location:	90 Alumni
Office:	127 Bell Hall
Office Hours:	WF: 10.00-11.30AM
Recitation R1:	M 8.00-8.50AM
Recitation R2:	F 5.00-5.50PM

Textbook and other material

The primary textbook for this course is:
Real-time Systems Design and Analysis by Philip Laplante, Third Edition, Wiley InterScience publishers, 2004.

While there are no other required textbooks, you should have in your possession appropriate reference books for both the C programming language.

We will also design embedded systems projects using Linksys WRT54G series wireless router that can embed Linux operating system. Specifically we will follow the projects discussed in the Embedded XINU project at http://xinu.mscs.mu.edu/

Pre-requisites

CSE341 Computer Architecture or permission of the instructor.

You will also be working on several large programming projects over the course of the semester. You must have a strong working knowledge of C (intermediate level background or above). This knowledge should extend to dynamic allocation and de-allocation of instances and fundamental pointer operations for class instances. Much of the lab work revolves around strong design, which you have all been exposed to since Programming/Computer Science I.

Grading Distribution

Grades will consist of the following components:

Component (Quantity)	Percentage
Labs (3)	15%, 20%, 20%
Midterm (1)	20%
Final (1)	25%

Point distribution guideline will be as follows:

Point Range	Letter Grade
95.00-100	A
90.00-94.99	A-
85.00-89.99	B+
80.00-84.99	B
75.00-79.99	B-
70.00-74.99	C+
65.00-69.99	C
60.00-64.99	C-
55.00-59.99	D+
50.00-54.99	D
0-49.99	F

I reserve the right to alter component weighting or provide a "curve" on an assignment as warranted.

NOTE : → COMPONENT PASS POLICY ← IN ORDER TO PASS THIS COURSE, YOU MUST HAVE PASSING WEIGHTED COMPONENT AVERAGES (WEIGHTED COMPONENT AVERAGES MUST BE GREATER THAN 49.99) THERE WILL BE TWO COMPONENTS THIS SEMESTER. COMPONENT 1 IS THE EXAM COMPONENT CONSISTING OF THE MIDTERM AND FINAL. COMPONENT 2 IS THE LAB COMPONENT CONSISTING OF THE THREE PROJECTS.

Labs

Lab assignments constitute a major portion of the course. Over the semester, you will be given three lab assignments. The lab experiments will cover fundamental areas of embedded systems. We will implement some of the labs using special hardware LinkSys WRT54GL. The labs will be based on Embedded XINU platform created at Marquette University. The students under the guidance of the instructor will work on the labs during the recitation hours. Currently there are 2 units of WRT54GL available for use by the students for completing the lab work.

Exams

There will be a Midterm that will be administered and graded before the resign date. Midterm material will cover all lecture and reading assignments before the exam, as well as concepts from the lab assignments. Midterms are closed book, closed notes, and closed neighbor. Please see the additional handout for exam taking policies for this course. The final is a comprehensive exam, covering all lecture, lab, and homework areas. The final is closed book, closed notes, and closed neighbor.

Attendance Policy

You are responsible for the contents of all lectures and recitations (your assigned section). If you know that you are going to miss a lecture or a recitation, have a reliable friend take notes for you. Of course, there is no excuse for missing due dates or exam days. We do, however, reserve the right to take attendance in both lecture and recitation. We may use this information to determine how to resolve borderline grades at the end of the course, especially if we see a lack of attendance and participation during lecture sessions. During lectures, we will be covering material from the textbook. We will also work out several of the problems from the text. Lecture will also consist of the exploration of several real world realtime and embedded systems problems not covered in the book. You will be given a reading assignment at the end of each lecture for the next class.

Office Hour Policy

If you can't meet during these hours, you will have to communicate with us via Email. Office hours are intended to resolve questions about the material that could not be answered in lecture or recitation. Come to office hours prepared.

Grading Policy

All assignments will be graded and returned in a timely manner. When an assignment is returned, you will have a period of one week to contest any portion of the grade. The TA who graded your assignment will be the first person to resolve a grading conflict. If the conflict cannot be resolved, the instructor will mediate the dispute. The judgment of the instructor will be final in all such cases. When contesting a grade, you must be able to demonstrate how your particular solution is correct. Also, when contesting a grade, the instructor or TA reserves the right to re-evaluate the entire lab or exam, not just the portion in dispute.

Incomplete Policy

We only grant incompletes in this course under the direst of circumstances. By definition, an incomplete is warranted if the student is capable of completing the course satisfactorily, but some traumatic event has interfered with their capability to finish within the timeframe of the semester. Incompletes are not designed as stalling tactic to defer a poor performance in a class.

Academic Integrity Policy

UB's definition of Academic Integrity in part is, "Students are responsible for the honest completion and representation of their work". It is required as part of this course that you read and understand the departmental academic integrity policy located at the following URL:
http://www.cse.buffalo.edu/academics-academic_integrity.shtml

There is a very fine line separating conversation pertaining to concepts and academic dishonesty. You are allowed to converse about general concepts, but in no way are you allowed to share code or have one person do the work for others. You must abide by the UB and Departmental Academic Integrity policy at all times. Remember that items taken from the Internet are also covered by the academic integrity policy! If you are unsure if a particular action violates the academic integrity policy, assume that it does until you receive clarification from the instructor. If you are caught violating the academic integrity policy, you will minimally receive a ZERO in the course.

Web Site

The CSE321 website should be checked frequently for important news. Course assignments, slides, grade reporting, and general hints and tips will be posted on the website.

Students with Disabilities

If you have special needs due to a disability, you must be registered with the Office of Disability Services(ODS). If you are registered with ODS please let your instructors know about this so that they can make special arrangements for you.

Week of	Topics Covered	Reading Material
8/27	General Introduction; Course Outline; Real-time definitions: time, events, determinism, synchronous and asynchronous events, CPU utilization; sample realtime systems; History of realtime systems	Ch.1
9/3	Introduction to Embedded XINU environment. Project 1 discussion:	http://xinu.mscs.mu.edu/
9/10	Hardware fundamentals: hardware interfacing: latching, edge versus level triggered, tristate logic, wait states, system interfaces and buses; MIL-STD-1553B, IEEE 1394 Firewire.	Ch.2; Sec 2.2
9/17	CPU hardware: Instructions, addressing, RISC vs CISC; Memory: Memory technologies: semiconductor memory, Ferriteelectric memory, memory hierarchy;	Ch.2: Sec 2.3, 2.4
9/24	Input/output: programmed input/output; direct memory access, memory-mapped IO, interrupts, interfacing devices to DPU via interrupts; enhancing performance through caching and pipelining; FPLAs; transducers;	Ch.2: Sec 2.5, 2.6, 2.7
10/1	Real-time operating systems: kernels, pseudo kernels, interrupt service routines, preemptive priority systems; task management in real-time operating systems.	Ch.2: Sec 2.4, 2.5
10/8	Theoretical foundations of real-time operating systems: process scheduling, task modeling, cyclic executive, fixed priority, dynamic priority; data buffers, buffer overflow, mailboxes, synchronization basics.	Ch.3. 3.1-3.4
10/15	Project 2 discussion	Project 2 handout
10/22	Realtime operating systems case studies: posix Pthread system ; clocks and timers and associated data structures;	Ch. 3: 3.5;
10/29,11/5	Embedded XINU: Embedded Linux; project 3 discussion	http://xinu.mscs.mu.edu/Downloads
11/12	Performance analysis: code execution time estimation, using system clock, analysis of co-routines, periodic, sporadic and aperiodic system analysis	Ch.7: 7.2
11/19	Faults, failures and bugs: testing: unit testing, black box and white box testing, debugging, coverage of test cases, system level testing, stress testing; fault tolerance	Ch.8: 8.2
11/26	Kalman filter, probing and patching, system integration; cost estimation	Ch.8: 4-8.6
12/3	Firmware and Rom resident execution strategies: case studies: wireless devices, digital signal processors	Class notes
	Review for the final exam	
Important Dates	**Due Date**	
10/4	Project 1	
11/2	Project 2	
12/4	Project 3	
10/11	Exam 1	
Finals Week	Final Exam	