Computer Science Department
College of Arts and Sciences

CAS CS350 / Spring 2000
Fundamentals of Computing Systems

Syllabus

Instructor:

Name: Prof. Azer Bestavros
Email: best@cs.bu.edu
Office: MCS-276 (ext: 3-9726)
Hours: Mon 12:30pm-2:00pm and Thu 5:00pm-6:30pm
or by appointment arranged through email

Teaching Fellow:

Name: Sumit Mehrotra
Email: sumit@cs.bu.edu
Office: MCS-288 (ext: 3-2065)
Hours: Thu 11:00am-1:00pm
or by appointment arranged through email

Lectures:

Place: SMG 105
Time: Tue 3:30pm-05:00pm / Thu 3:30pm-05:00pm

Sections:

Place: CS Undergraduate Laboratory
Time: Tue 12:00-1:00/2:00-3:00/6:00-7:00/7:00-8:00/8:00-9:00pm
(students attend one of the above sessions each week)

Cyberspace Coordinates:

URL: http://www.cs.bu.edu/~best/crs/cs350/S00
Email: cs350@cs.bu.edu
(To add yourself type "**csmail -a cs350**" from your csa account)

Overview

This course is a required junior-level Computer Science course. It covers the fundamental concepts underlying the design and implementation of computing systems.

Purpose of Course:
The philosophy underlying the design of this course is that students should be familiarized with problems that reoccur in software systems, and should be acquainted with the set of classical algorithms/techniques for solving such problems. In particular, it is important to develop the ability to recognize standard problems in different wordings and within unusual context, and match them with appropriate solutions.

Catalog Description:
Rigorous treatment of invariant concepts, algorithms, and performance evaluation methods underlying computing systems design. Topics include modeling and analysis of concurrent processing, computational resource scheduling and consumption, and performance evaluation techniques.

Prerequisites:
CS-210: Basic computer organization and software/hardware interface concepts, which are covered in CS-210.
CS-113: Basic algorithms and data structures, which are covered in CS-112 or CS-113.
MA-293: Basic mathematical structures and elementary probability theory which are covered in MA-293.

Function in the Curriculum:
As a core undergraduate course, CS-350 does not focus on particular implementations or specific technologies. On the contrary, it stresses the fundamental concepts and basic algorithms that have survived (and are likely to survive) the evolution of computer software systems in general, and operating systems in particular.

This course is significantly different in purpose and coverage from CS-210 (Computer Systems). CS-210 introduces computer systems to sophomores using a hands-on approach by examining the "mechanics" of modern (or typical) computer operation through exposure to various interfaces between architecture, compilers, loaders, linkers, and run-time systems. CS-350 does not focus on a particular interface or a particular technology, rather it deals with fundamental notions and algorithms that are common to computing systems in general and to software systems in particular.

This course is required for the following advanced courses: CS-552 (Advanced Operating Systems), CS-550 (Advanced Computer Architecture), CS-551 (Parallel Computing Architectures and Models), CS-555 (Data Communications), CS-560 (Databases), and CS 470/670 (Performance Analysis). In that respect, this course covers basic notions that we expect students (graduates and undergraduates alike) enrolling into these classes to have mastered. For example, it is inconceivable that a student who is interested in taking a database or an operating systems class, not to have been exposed to the notion of concurrent processing and mutual exclusion, for example. Similarily, it is unconceivable for a student enrolled in a networking, database, or operating systems class, not to have been exposed to performance analysis using simple queueing systems, for example.

Topical contents:

The topics covered in this class could be grouped under 3 general themes:

- Performance Analysis and Evaluation
- Scheduling and Resource Management
- Concurrency and Synchronization

The first half of the semester will be devoted to the coverage of the first two of the above themes, whereas the second half of the semester will be devoted to the coverage of the third of the above themes.

It is very important to realize that this is NOT an Operating Systems course. Despite the nature of the textbook, this course will not get into specifics of operating systems (e.g. API, Unix/Solaris/NT implementations). Of course, the concepts studied in this course are central to more advanced system courses (including Operating Systems, Networking, and Databases, as discussed above).

Textbooks and Class Notes

Textbooks and References:

- [S] William Stallings, Operating Systems: Internals and Design Principles, 3/e. Published January 1998 by Prentice Hall ESM. 781 pp. ISBN 0-13-887407-7. (required)
  Note: This book is available from the BU Bookstore. Also, you can purchase this book from Amazon.com (price = $71.00).

- [W] Azer Bestavros, Lecture Notes on Fundamentals of Computing Systems. These notes are work-in-progress. They complement the textbook and, thus, constitute an integral part of the required readings. (required)
  Note: These Lecture Notes will be available from theClass Home Page.

- [A] Arnold Allen, Probability, Statistics, and Queueing Theory with Computer Science Applications, 2/e, Academic Press, hardcover, 740 pages, 1990., ISBN 0-12-051051-0 (optional)
  Note: This book is available from the BU Bookstore. Also, you can purchase this book from Amazon.com (price = $48.00).

Lectures, Discussion Sections, and Other Channels

Course Lectures:
Attending lectures is mandatory and will account for 10% of the final class grade. To take attendance, frequent one-question quizzes (at the end of a randomly selected set of lectures) will be given.
You are responsible for ALL the materials covered in the lectures *including any topics not in the textbooks.*

Class participation and questions are very welcome during the lectures. There is no such thing as a ``stupid question''. Failure to ask a simple question might result in the inability to follow the rest of the lecture.

Up to 10 minutes at the beginning of every class will be devoted to answering questions related to topics discussed in previous lectures. This is a particularly good time to ask if you have questions.

Course Discussion Sections:
If you are taking this class then you have signed up for a weekly one-hour discussion session. The discussion session will be conducted in the Computer Science Undergraduate Laboratory. The Teaching Fellow for the clas is Sumit Mehrotra.

The teaching fellow will be leading the discussion sessions. Materials covered in the discussion sessions will be elaborations on materials covered in the lectures. Also, the discussion sessions will be the venue used to answer any questions (or provide clarifications) regarding the homework assignments.

If (for some reason) you miss (or cannot make) the discussion section for which you are signed up, then please make sure to attend a later one during the same day.

Office Hours:
You should come to the Instructor and/or Teaching Fellow's office hours with your questions well-thought out. Office hours are NOT meant to be tutoring sessions; they are meant to answer specific questions about the material covered in the lecture, discussion section, and textbook(s). Questions like "could you repeat your explanation of ..." or "I do not understand section ... of the textbook" should be asked in the lecture and/or discussion sections.

Mailing List Usage:
The class mailing list is for you to use as a channel for asking questions and seeking clarifications, whether from the instructor, teaching fellow, or classmates. However, like any broadcast medium, you should be careful not to abuse that list. This means that "one-liners" and "cute" commentaries should be avoided. Also, note that you will be held responsible for any inappropriate emails sent on this list.

Homework Assignments

Homework assignments constitute an important part of this course. They are designed to help you understand the materials covered in lectures and in assigned readings. It is only by doing the homework that you really learn the material. There will be approximately 6-8 homework assignments handed out throughout the semester. Homework Assignments will be worth 30% of the final grade. Homework assignments will NOT be weighted equally. Harder/longer homework assignments will constitute a larger percent of the combined grade for all homeworks.

Some of the the problems in the homework may require minor programming or use of special educational software to write concurrent software. One of the homework assignments will require writing a discrete event simulator for a queuing system, which (for some) may prove a bit challenging given the ``different" nature of such a program (management of queues, events, state of the queuing system and of the simulation, etc.)

Homework assignments may be fairly long, each taking an average of 3 to 4 hours to complete, although the time required will vary quite a bit from student to student and assignment to assignment. Always allow more time than you think you will need!

Homework assignments will be handed out at the end of class meetings. Extra copies will be left in one of the handouts box, labeled ``CS-350 / Handouts", in the hall outside of MCS 137. Also, an on-line copy will be made available through the Course Web Page.

Homework assignments are to be handed in using the slotted homework box, labeled ``CS-350 / Drop Box", in the hall outside of MCS 137 by 5:00 pm on the day they are due. Late homework must be time-stamped by a CS Office staff and left in Sumit Mehrotra's (the Teaching Fellow's) mailbox (NEITHER in the Drop Box NOR in my Mailbox). Do not hand in your homework in the class or during office hours. Do not hand in your homework by slipping them under the office door of the instructor.

There will be a hefty penalty of 25% for a homework handed in one class late, and of 50% for a homework handed in two classes (i.e. one week) late. No homework will be accepted if late for more than one week. There will be NO exceptions to this policy, other than for religious holidays and certified medical excuses. In such cases, extensions will be granted only if (and until) the homework solutions are posted (hopefully, about 1 week after the original due date).

Graded assignments will be returned in a timely fashion. They will be available for pick-up during section hours. It is the student's responsibility to make sure that the grade they received for each homework is properly recorded by the instructor and/or teaching fellow. This can be done by periodically requesting on-line grade reports. Please check the course's home page for instructions on how to request such reports. Discrepancies will be corrected as long as the student can show the graded homework with the correct grade on it. Thus, make sure to get back your graded assignments.

If you believe that there is a chance that your homework assignment will be lost by the course staff, then here is how you can protect yourself: (1) Make a copy of your homework assignment before handing it in, and (2) Have the Computer Science Department office time-stamp your copy of the homework assignment. Claims for ``lost" assignments will be considered only if accompanied by a time-stamped copy of what you handed in. There are no exceptions to this rule.

Quizzes and Exams

Attendance Quizzes:
Random one-question quizzes will be administered throughout the semester. Each of these quizes will be given at the end of a lecture and will each consist of a single (very easy) question on the material covered in that lecture. The main purpose of these quizzes is to be a measure of attendance, since I expect anybody who attended the class (and was not asleep) to be able to answer the question. Attendance Quizes will be worth 10% of your grade.

Midterm Exam:
There will be an in-class midterm exam. This exam will be worth 20% of the final grade, and will cover the material presented from the beginning of the semester until the week before the midterm exam. Please check the class schedule for time and place.

Final Exam:
The course final exam is worth 40% of the final grade, and will cover the material offered THROUGHOUT the semester. Please check the class schedule for date and time. The place for the final exam will be announced in due time.

Policy on Missed Exams:
Please mark the exam dates on your calendar (and remember them when you make your recess and end-of-semester travel plans!) There will be ABSOLUTELY NO make-up exams, except for medical emergencies. In that respect, I will not accept blue slips from Health Services; you must justify your medical problem with a letter from a doctor, specifying the period of time during which you were unable to attend one of the two exams.

Grading Policies

Grading:
Grading (except for the final exam) is done by a number of class graders, under the direct supervision of the Teaching Fellow and the Instructor. If you have an issue with a grade (homework or exam), please contact the Teaching Fellow. If your issue is not resolved, then (and only then) please contact the Instructor. In doing so, please note that (to ensure fairness and grading consistency) it is seldom the case that the Instructor will overrule the Teaching Fellow.

This class is NOT graded on a curve. This means that if everybody in the class is an A-student, then everybody will get an A. The converse is also true! Therefore, don't be satisfied with an "average" grade because that average could well be less than what you expect! In previous years, the average grade for this class was between a C+ and a B-, which is low compared to other classes in CS.

Being a "gate" to all other (more advanced) CS "systems" classes (e.g. CS550, CS552, CS555, CS560, etc.), you should expect this class to be competitive. Thus, make sure you work hard from the very beginning!

Grade Breakdown:
The final class grade will be broken down as follows:

- 30% on Homeworks
- 10% on Attendance Quizzes
- 20% on Midterm Exam
- 40% on Final Exam

About the Instructor

Azer Bestavros received his S.M. and Ph.D. degrees from Harvard University in 1988 and 1992, respectively. He is an Associate Professor of Computer Science at Boston University, where he conducts research on Networking, Internet/Web Protocols and Systems, and Real-Time Communication and Systems. Professor Bestavros has co-edited three books and has authored or co-authored in excess of 70 refereed technical publications. He is currently the editor-in-chief of the Newsletter of the IEEE-CS TC on Real-Time Systems and the General Chair of the IEEE Real-Time Technology and Application Symposium. Professor Bestavros' past course offerings included Introduction to Computer Science, Fundamentals of Computing Systems, the introductory and advanced Computer Architecture courses, in addition to various seminars on Real-Time and Networked Information Systems. Professor Bestavros' research is funded through grants from the National Science Foundation, the Army Research Office, GTE, and Microsoft. Professor Bestavros' professional activities include his consulting work for a number of industrial labs and Internet startups.

From Collaboration to Plagiarism

Academic Code of Conduct:
In addition to the normal and well-understood strictures against cheating on exams, altering transcripts, and etc., there are other varieties of academic misconduct described in the Academic Conduct Code of Boston University which you must be aware of when working on assignments. The most relevant sections (B, F, and I) of the Academic Code for the assignments in this class are as follows:

- Plagiarism. Any attempt by a student to represent the work of another as his or her own. This includes copying the answer of another student on an examination or copying or substantially restating the work of another person or persons in any oral or written work without citing the appropriate source, and collaborating with someone else in an academic endeavor without acknowledging his or her contribution.
- Knowingly allowing another student to represent your work as his or her own.
- Submitting the same work in more than one course without the consent of the instructors involved.

Policy on Collaboration:
There is nothing wrong in principle about discussing the topics covered in the course with your friends and colleagues. However, there are severe consequences to *plagiarism*. In particular, when you submit a homework with your name on it, you are claiming that the work contained therein is your own; it is *Plagiarism* to submit work under your own name in which:

- You collaborated with another student (current, former, friend, etc.) in solving homework problems;
- You copied the solution from another student; or
- You obtained the solution from a book or other sources.

The above is nothing less than *plagiarism* and will be punished accordingly.

If you are in a situation whereby you have collaborated with a student on a homework problem, or obtained homework solutions (or parts thereof) from some other source, then you should CLEARLY and UNAMBIGUOUSLY disclose this in your writeup. Failure to do so constitutes plagiarism.

If you are caught cheating in a test or plagiarising a homework, you will AUTOMATICALLY receive an F and the matter will be reported to CAS Academic Conduct Committee, who recommends sanctions to the Dean of the College. Handing in your own assignment a day or two late will affect your grade FAR LESS than turning in a copy of someone else's work on time!

CAS CS350 / Spring 2000

Schedule

Date	Topic	Readings
1/11	Review of Computer System Organization	[S:1,2]
1/13	The Concept of a Process, Process State and Process State Transitions	[S:3.1-3.3]
1/20	Performance Metrics of Computing Systems	[W]
1/25	Elementary Queuing Analysis: Review of elementary probability	[S:A], [W]
1/27	Elementary Queuing Analysis: The M/M/1 queuing model	[S:A], [W]
2/01	Elementary Queuing Analysis: Queuing Networks	[S:A], [W]
2/03	Elementary Queuing Analysis: Case Study	[W]
2/08	Empirical Performance Analysis: Discrete Event Simulation	[W]
2/10	Empirical Performance Analysis: Measurements, Sampling and Estimation	[W]
2/15	Empirical Performance Analysis: Building Confidence in Metric Estimates	[W]
2/17	Resource Management: Scheduling Fundamentals	[S:9.1]
2/24	Resource Management: Basic Scheduling Algorithms	[S:9.2]
2/29	Resource Management: Real-Time and Priority-Based Scheduling Algorithms	[S:10.2] [W]
3/02	Resource Management: State-Sensitive (I/O) Scheduling Algorithms	[S:11.5]
3/14	Midterm Exam (3:30pm-5:00pm in SMG-105)	
3/16	Principles of Concurrency and Process Synchronization	[S:5.1]
3/21	Mutual Exclusion: Dekker's Algorithm	[S:5.2]
3/23	Mutual Exclusion: Peterson's Algorithm	[S:5.2]
3/28	Mutual Exclusion: Semaphores and Producer/Consumer Problems	[S:5.4]
3/30	Mutual Exclusion: Semaphore Implementation	[S:5.4]
4/04	Classical Synchronization Problems: The Barbershop Problem	[S:5.4]
4/06	Classical Synchronization Problems: Deadlock Management	[S:6.1-6.4]
4/11	Classical Synchronization Problems: The Dining Philosophers Problem	[S:6.6]
4/13	Classical Synchronization Problems: Transactions and Serializability	[W]
4/18	Classical Synchronization Problems: Logging and Recovery	[W]
4/20	Distributed Algorithms: The Distributed Global Snapshot Problem	[S:13,14.2]
4/25	Distributed Algorithms: The Clock Synchronization Problem	[S:13,14.2]
4/27	Course Conclusion: Review of Performance Evaluation and Scheduling	
5/02	Course Conclusion: Review of Concurrency Management	
5/11	Final Exam (9:00am-11:00am)	

Updates to this schedule will be announced and posted if need be.