Boston University
College of Arts and Sciences
Computer Science Department

A. Bestavros, 1991

111 Cummington Street, MCS
Boston, Massachusetts 02215
Web Home Page at http://www.cs.bu.edu
Tel: (617) 353-8919 / Fax: (617) 353-6457

CAS CS-350 / Spring 1998

Fundamentals of Computing Systems

http://www.cs.bu.edu/faculty/best/crs/cs350/S98

Tue & Thu: 9:30am-11:00am
MCS-149

Course Instructor: Prof. Azer Bestavros

- **Office:** MCS 276
- **Email:** best@bu.edu
- **Web:** http://www.cs.bu.edu/faculty/best
- **Phone:** 353-9726
- **Office Hours:** Thu 11:00am–12:30pm, or by appointment

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:

This course assumes that students are familiar with:

a. Basic computer organization and basic system design concepts such as the abstract layering of computing systems as concentric spheres of interpretation and control. This is covered in CS-210 (Computer Systems).
b. Basic algorithms and data structures, which are covered in CS-112 (or CS-113).
c. Basic mathematical structures and elementary probability theory which are covered in MA-293.

Overview:

This course is a required junior-level Computer Science course. It covers the fundamental concepts underlying the design and implementation of computing systems. This includes the concepts of (1) process, process state and transitions, (2) concurrent processing and synchronization, (3) scheduling theory and resource management algorithms, and (4) analytical and empirical performance evaluation techniques.

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 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 algorithms.

Spring 1998 marks the first time this course is offered. Thus, students are forewarned that the schedule for the course is untested, in the sense that the length of time it will take to cover specific topics may differ (hopefully, not significantly) from the schedule at the end of this handout.

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 will be 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), and CS-560 (Databases). 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 class, not to have being exposed to the notion of concurrent processing and mutual exclusion, for example.

Textbooks:

Because the following textbooks do not follow the philosophy behind CS-350's conception closely enough, they will be supplemented with extensive notes and problem sets.

[**S**] William Stallings, Operating Systems, 3/e, Published January 1998, by Prentice Hall ESM. 781 pp. ISBN 0-13-887407-7. `http://www.shore.net/ ws/OS3e.html` *This book is the required textbook for the course.*

[**W**] *Web Resources.* A collection of on-line lecture notes and reading materials will be available from the Course Home Web Page. The materials on these Web Pages will complement the textbook and, thus, constitute an integral part of the required readings.

[**A**] Arnold Allen, Probability, Statistics, and Queueing Theory with Computer Science Applications, 2/e, Academic Press, hardcover, 740 pages, 1990. *This book is an optional reference.*

Topical Coverage:

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 <u>NOT</u> 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).

Lectures:

- 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.

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 40% of the final grade.
- Some of the the problems in the homework may require minor programming or use of special educational software to write concurrent software.
- 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 / Outbox", in the hall outside of MCS 137.
- Homework assignments are to be handed in using the slotted homework box, labeled "CS-350 / Inbox", 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 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).

Exams:

- There will be **an in-class midterm exam** on Tuesday, March 17^{th}, from 9:30am to 11:00am. This exam will be worth 20% of the final grade, and will cover the material presented from the beginning of the semester and until March 5.
- There will be a **final exam** on Tuesday, May 5^{th} from 9:00am to 11:00am. This exam will be worth 40% of the final grade, and will cover the material offered *throughout* the semester.
- Please mark the exam dates on your calendar (and remember them when you make your spring break 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.

Course Grade:

- The course grade will be broken down as follows:
  - **40%** Homework assignments
  - **20%** Midterm exam on 3/17/1998
  - **40%** Final exam on 5/5/1998
- There will be **no incompletes** in this course except for reasons of dire illness shortly before the end of the course, and only if a significant amount of work has been completed (*e.g.* attending lectures, handing in most homework assignments, and attending the midterm).

Academic Code of Conduct and Policy on Collaboration:

- 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.*
- 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 copying their solutions. This is nothing less than **plagiarism** and will be punished accordingly. If you are caught cheating in a test or a homework, you will **automatically receive an F** and the matter may 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!

About the Instructor:

Azer Bestavros received his S.M. and Ph.D. degrees from Harvard University in 1988 and 1992, respectively. He is a Computer Science faculty member at Boston University, where he conducts research on Real-Time Computation and Communication Systems and on Large-Scale Networked Information Systems. Professor Bestavros has co-edited two books on real-time databases and has authored or co-authored in excess of 60 refereed technical publications. He is currently the editor-in-chief of the Newsletter of the IEEE-CS TC on Real-Time Systems and the PC Chair of the IEEE Real-Time Technology and Application Symposium. Professor Bestavros' course offerings include Introduction to Computer Science (CS-101), Fundamentals of Computing Systems (CS-350), the introductory (CS-450) and advanced (CS-550 and CS-551) Computer Architecture courses, in addition to various seminars on Real-Time and Networked Information Systems. Prof. Bestavros is the Faculty Advisor of the ACM Student Chapter at BU. He is a member of BU's Center for Computational Science. He chairs the College's Natural Science Curriculum Committee and serves on the University's Educational Technologies Committee.

Boston University
College of Arts and Sciences
Computer Science Department

111 Cummington Street, MCS
Boston, Massachusetts 02215
Web Home Page at http://www.cs.bu.edu
Tel: (617) 353-8919 / Fax: (617) 353-6457

CAS CS-350 / Spring 1998
Fundamentals of Computing Systems

Syllabus

Class	Topic	Readings
1/13	Review of Computer System Organization	[S:1,2]
1/15	The Concept of a Process, Process State and Process State Transitions	[S:3.1]
1/20	Process Description and Control	[S:3.2-3.3]
1/22	Performance Metrics of Computing Systems	[W]
1/27	Elementary Queuing Theory: Review of elementary probability	[S:A]
1/29	Elementary Queuing Theory: The M/M/1 queuing model	[S:A]
2/03	Elementary Queuing Theory: Queuing Networks	[S:A]
2/05	Elementary Statistical Analysis: Sampling and Estimation	[W]
2/10	Elementary Statistical Analysis: Confidence Intervals	[W]
2/19	Scheduling Fundamentals	[S:9.1]
2/24	Classical Scheduling Algorithms	[S:9.2]
2/26	Real-Time and Priority-Based Scheduling Algorithms	[W][S:10.2]
3/03	I/O Scheduling Algorithms	[S:11.5]
3/05	Slack (Conclusion and Review)	
3/17	Midterm Exam in MCS-149 (in class from 9:30am to 11:00am)	
3/19	Principles of Concurrency and Process Synchronization	[S:5.1]
3/24	Mutual Exclusion: Dekker's Algorithm	[S:5.2]
3/26	Mutual Exclusion: Peterson's Algorithm	[S:5.2]
3/31	Mutual Exclusion: Semaphores and Producer/Consumer Problems	[S:5.4]
4/02	Mutual Exclusion: Semaphore Implementation	[S:5.4]
4/07	Deadlock: Characterization and Prevention	[S:6.1-6.2]
4/09	Deadlock: Avoidance and Detection	[S:6.3-6.4]
4/14	Classical Synchronization Problems: The Barbershop Problem	[S:5.4]
4/16	Classical Synchronization Problems: The Dining Philosophers Problem	[S:6.6]
4/21	Classical Synchronization Problems: Transactions and Serializability	[W]
4/23	Distributed Computing and Algorithms	[S:13]
4/28	Distributed Computing and Algorithms	[S:14.2]
4/30	Slack (Course conclusion, evaluation, and review)	
5/05	Final Exam in MCS-149 (from 9:00am to 11:00am)	

** Changes to this schedule will be announced if need be.*