

## Course Information

### Staff

#### *Faculty*

|                 |          |         |                         |
|-----------------|----------|---------|-------------------------|
| Butler Lampson  | NE43-535 | x3-6004 | blampson@microsoft.com  |
| Alex Shvartsman | NE43-371 | x3-6182 | alex@theory.lcs.mit.edu |

#### *Teaching Assistants*

|                |          |         |                           |
|----------------|----------|---------|---------------------------|
| Mandana Vaziri | NE43-369 | x3-6097 | vaziri@theory.lcs.mit.edu |
|----------------|----------|---------|---------------------------|

#### *Course Secretary*

|               |          |         |                           |
|---------------|----------|---------|---------------------------|
| Joanne Talbot | NE43-366 | x3-6054 | joanne@theory.lcs.mit.edu |
|---------------|----------|---------|---------------------------|

#### *Office Hours*

Messrs. Lampson and Shvartsman will arrange individual appointments. Mandana Vaziri will hold scheduled office hours outside of NE43-365. In addition to holding regularly scheduled office hours, the TA will also be available by appointment.

### Lectures

Lectures are held on Mondays and Wednesdays from 1:00 to 2:30PM in room 38-166. Messrs. Lampson and Shvartsman will split the lectures.

### Handouts

The source material for this course will be an extensive set of handouts. There are about 300 pages of topic handouts that take the place of a textbook; you will need to study this material to do well in the course. Seven research papers supplement the topic handouts. In addition there are 12 problem sets. Solutions for each problem set will be available shortly after the due date.

There is a course Web page, at URL <ftp://theory.lcs.mit.edu/pub/classes/6.826/www/6.826-top.html>. Last year's handouts can be found from this page. Current handouts will be placed on the Web as they are produced, as Postscript (.ps), Acrobat (.pdf), and Word (.rtf) files.

Current handouts will generally be available in lecture. If you miss any in lecture, you can obtain them afterwards from the course secretary. She keeps them in a file cabinet outside her office.

## **Problem sets**

There is a problem set approximately once a week. Problem sets are handed out on Wednesdays and are due in class the following Wednesday. They normally cover the material discussed in class during the week they are handed out. Delayed submission of the solutions will be penalized, and no solutions will be accepted after Thursday 5:00PM.

Students in the class will be asked to help grade the problem sets. Each week a team of students will work with the TA to grade the week's problems. This takes about 3-4 hours. Each student will probably only have to do it once during the term.

We will try to return the graded problem sets, with solutions, within a week after their due date.

### *Policy on collaboration*

We encourage discussion of the issues in the lectures, readings, and problem sets. However, if you collaborate on problem sets, you must tell us who your collaborators are. And in any case, you must write up all solutions on your own.

## **Grades**

There is a final exam, but no midterm. Grades are based 60% on the problem sets, 30% on the final exam, and 10% on quality and promptness of grading.

## **Course mailing list**

A mailing list for course announcements—6826@mit.edu—has been set up to include all students and the TA. If you do not receive any email from this mailing list within the first week, check with the TA. Two additional mailing lists have also been provided: 6826-staff@mit.edu allows students to send email to the entire 6.826 staff, and 6826-disc@mit.edu is an unofficial forum for discussion among students.

## Course Schedule

| Date              | No | Who | HO | Topic   | PS out | PS due |
|-------------------|----|-----|----|---|--------|--------|
| Wed.,<br>Feb. 5   | 1  | L   |    | <b>Overview.</b> The Spec language. State machine semantics.<br>Examples of specifications and implementations.<br>1 Course information<br>2 Background<br>3 Introduction to Spec<br>4 Spec reference manual<br>5 Examples of specs and implementations                 | 1      |        |
| Mon.,<br>Feb. 10  | 2  | S   |    | <b>Specification and implementation</b> for sequential programs.<br>Correctness notions and proofs. Proof methods: abstraction<br>functions and invariants.,<br>6 Abstraction functions   |        |        |
| Wed.,<br>Feb. 12  | 3  | L   |    | <b>File systems 1:</b> Disks, simple sequential file system, caching,<br>logs for crash recovery.<br>7 Disks and file systems   | 2      | 1      |
| Tues.,<br>Feb. 18 | 4  | S   |    | <b>File systems 2:</b> Copying file system.   |        |        |
| Wed.,<br>Feb. 19  | 5  | S   | 3  | <b>Proof methods:</b> history and prophecy variables; abstraction<br>relations.<br>8 History variables  | 3      | 2      |
| Mon.,<br>Feb. 24  | 6  | L   |    | <b>Semantics and proofs:</b> Formal sequential semantics of Spec.<br>9 Atomic semantics of Spec   |        |        |
| Wed.,<br>Feb. 26  | 7  | S   |    | <b>Performance:</b> how to get it, how to analyze it.<br>10 Performance<br>11 Paper: Michael D. Schroeder and Michael Burrows,<br>Performance of Firefly RPC, <i>ACM Transactions on Computer<br/>           Systems</i> <b>8</b> , 1, February 1990, pp 1-17.          | 4      | 3      |
| Mon.,<br>Mar. 3.  | 8  | L   |    | <b>Naming:</b> specs, variations, and examples of hierarchical<br>naming.<br>12 Naming<br>13 Paper: David Gifford et al, Semantic file systems, <i>Proceedings<br/>           of the 13th ACM Symposium on Operating System Principles</i> ,<br>October 1991, pp 16-25. |        |        |
| Wed.,<br>Mar. 5   | 9  | L   |    | <b>Concurrency 1:</b> Practical concurrency, easy and hard. Easy<br>concurrency using locks and condition variables. Problems<br>with easy concurrency: scheduling, deadlock.   | 5      | 4      |

| Date             | No | Who | HO | Topic   | PS out | PS due |
|------------------|----|-----|----|---|--------|--------|
|                  |    |     |    | 14 Practical concurrency  |        |        |
|                  |    |     |    | 15 Paper: Andrew Birrell, An Introduction to Programming with Threads, Digital Equipment Corporation Systems Research Center Report 35, January 1989.   |        |        |
| Mon.,<br>Mar. 10 | 10 | L   |    | <b>Concurrency 2:</b> Concurrency in Spec: threads and non-atomic semantics. Big atomic actions. Safety and liveness. Examples of concurrency.  |        |        |
|                  |    |     |    | 16 Formal concurrency   |        |        |
| Wed.,<br>Mar. 12 | 11 | S   |    | <b>Concurrency 3:</b> More examples. Proving correctness of concurrent programs.  | 6      | 5      |
| Mon.,<br>Mar. 17 | 12 | S   |    | <b>Concurrency 4:</b> Examples of correctness proofs  |        |        |
| Wed.,<br>Mar. 19 | 13 | L   |    | <b>Concurrent systems:</b> Disks and file systems.  | 7      | 6      |
|                  |    |     |    | 17 Concurrent disks   |        |        |
| Mar.<br>22-30    |    |     |    | Spring Vacation   |        |        |
| Mon.,<br>Mar. 31 | 14 | L   |    | <b>Sequential transactions</b> with caching.  |        |        |
|                  |    |     |    | 18 Sequential transactions  |        |        |
| Wed.,<br>Apr. 2  | 15 | L   |    | <b>Concurrent transactions:</b> Specs for serializability. Ways to implement the specs.   | 8      | 7      |
|                  |    |     |    | 19 Concurrent transactions  |        |        |
| Mon.,<br>Apr. 7  | 16 | S   |    | <b>Introduction to distributed systems:</b> Characteristics of distributed systems. ISO Reference Model: physical, data link, and network layers. Design principles.  |        |        |
|                  |    |     |    | 20 Distributed systems  |        |        |
|                  |    |     |    | 21 Paper: Michael Schroeder et al, Autonet: A high-speed, self-configuring local area network using point-to-point links, <i>IEEE Journal of Selected Areas in Communications</i> 9, 8, October 1991, pp 1318-1335. |        |        |
| Wed.,<br>Apr. 9  | 17 | S   |    | <b>Networks 1:</b> Links. Point-to-point and broadcast networks. Ethernet. Token Rings.   | 9      | 8      |
|                  |    |     |    | 22 Networks: Links and switches   |        |        |
| Mon.,<br>Apr. 14 | 18 | S   |    | <b>Networks 2:</b> Switches. Implementing switches. Routing. Learning topologies and establishing routes.   |        |        |

| Date             | No | Who | HO | Topic  | PS out | PS due |
|------------------|----|-----|----|--|--------|--------|
| Wed.,<br>Apr. 16 | 19 | S   |    | <b>Networks 3:</b> Network objects and remote procedure call (RPC).<br>23 Network objects<br>24 Paper: Andrew Birrell et al., Network objects, <i>Proceedings of the 14th ACM Symposium on Operating Systems Principles</i> , Asheville, NC, December 1993.  | 10     | 9      |
| Mon.,<br>Apr. 21 |    |     |    | Patriot's Day, no class.   |        |        |
| Wed.,<br>Apr. 23 | 20 | L   |    | <b>Networks 4:</b> Reliable messages. 3-way handshake and clock implementations. TCP as a form of reliable messages.<br>25 Paper: Butler Lampson, Reliable messages and connection establishment. In <i>Distributed Systems</i> , ed. S. Mullender, Addison-Wesley, 1993, pp 251-281.                          |        | 10     |
| Mon.,<br>Apr. 28 | 21 | L   |    | <b>Distributed consensus</b> in the presence of faults. Paxos algorithm for asynchronous consensus.<br>26 Consensus  |        |        |
| Wed.,<br>Apr. 30 | 22 | L   |    | <b>Distributed transactions:</b> Commit as a consensus problem. Two-phase commit. Optimizations.<br>27 Distributed transactions  | 11     |        |
| Mon.,<br>May 5   | 23 | S   |    | <b>Replication and availability:</b> Implementing replicated state machines using consensus. Applications to replicated storage.<br>28 Replication<br>29 Paper: Jim Gray and Andreas Reuter, Fault tolerance, in <i>Transaction Processing: Concepts and Techniques</i> , Morgan Kaufmann, 1993, pages 93-156. |        |        |
| Wed.,<br>May 7   | 24 | S   |    | <b>Caching 1:</b> Maintaining coherent memory. Broadcast (snoopy) and directory protocols. Examples: multiprocessors, distributed shared memory, distributed file systems.<br>30 Concurrent caching  | 12     | 11     |
| Mon.,<br>May 12  | 25 | S   |    | <b>Caching 2.</b>  |        |        |
| Wed.,<br>May 14  | 26 | L   |    | <b>Reserve</b>   |        | 12     |
| May 20-24        |    |     |    | Finals week  |        |        |