Course Information and Standards
OPERATING SYSTEMS I
91.515

University of Massachusetts Lowell
Department of Computer Science
September, 2011

Instructor: Prof W. Moloney
Office: Olsen-222
Telephone: 978-934-3640
Office Hours: MWF 11:00 - 12:00 W 1:00-2:00 (other hours by appt.)
Email: bill@cs.uml.edu
Web: www.cs.uml.edu/~bill/cs515

1. Course description

A functional level view of multiprocessing operating systems including processor, memory, peripheral, and file systems management in batch, timesharing, real time, and distributed systems targeted for various hardware. Several programming assignments will be given for implementation on a UNIX/Linux platform (other OS platforms are negotiable, but in-class examples will be targeted for UNIX/Linux and structured in 'C').

Prerequisites: 91.308 or equivalent, and programming skill with 'C' or 'C++'

The above is the course description as it appears in the Graduate School Catalogue. In this course, particular emphasis will be placed on the problems of concurrency control in both centralized and distributed systems. It is assumed that the student has a working knowledge of operating systems at the level of an undergraduate course; should this not be the case, a text such as one of the following is suggested for background knowledge.

Avi Silberschatz , Peter Glavin and Greg Gagne.
Operating System Concepts, 8th Ed.
John Wiley & Sons, 2009.
(ISBN 978-0-470-12872-5)

William Stallings.
Operating Systems, 6th Ed.
Pearson Education, 2009
(ISBN 978-0136006329).

Andrew S. Tanenbaum.
Modern Operating Systems. 3rd Ed.
Pearson Education, 2008.
(ISBN 0136006639)

Gary Nutt
Operating Systems, 3rd Ed.
Addison-Wesley, 2004.
(ISBN 0201773449)

Deitel, Deitel and Choffnes
Operating Systems, 3rd Ed.
Pearson Education, 2004
(ISBN 0131828274)

2. Required Textbook

Mukesh Sinhgal and Niranjan Shivaratri.
Advanced Concepts in Operating Systems.
McGraw-Hill, Inc., 1994
(ISBN 007057572X)

Readings in the text will be supplemented from time to time by assigned readings of classic papers in the field; copies of these papers will be available through sources discussed in class.

3. Tests and Grading

In addition to the assigned readings, there will be two exams and several programming assignments. The proportions of your cumulative grade will be as follows:

Full class Exam: First Half Material	30%
Full class Exam: Second Half Material	30%
Assigned Programming Projects	40%

The first exam will cover the first half of the course and the second will cover only the second half of the course. All exams will be closed book and notes. See the attached schedule for dates.

4. Assignments

Regular assignments will include readings in the course textbook and assigned papers. A detailed description of the programming projects will be supplied separately.

5. Academic Dishonesty

In this course all work is to be each student's own. Students should therefore be familiar with the University's rules on academic dishonesty, which can be found in the *Graduate School Catalogue* and in the *Schedule of Classes*. In particular, ***plagiarism will not be tolerated!*** Any student caught plagiarizing another's work will automatically receive a grade of **F** for the course. If you are unsure as to what constitutes plagiarism, it is your responsibility to check with the instructor. Other forms of dishonesty will result in similar actions. Submission of **shared student code** is not permissible, and will result in a grade of **F** for the course. Help files are typically provided for each programming assignment, and students are encouraged to cut and paste useful code from

these help files into their assignment submissions, but all other code must be the sole and specific work of each individual student.

6. Course Outline

1. Process Synchronization [Sept 1, 8] [Chapter 2]
    - The Process Model
    - Competing and Cooperating Processes
    - Hardware Approach to Synchronization
    - Semaphores
    - General Events
    - Eventcounts and Sequencers
    - Language Based Synchronization Mechanisms
    - Monitors
    - Serializers
    - Path Expressions
    - The Ada Rendezvous Mechanism
    - Verification of Parallel Programs

2. Deadlock [Sept 15, 22] [Chapter 3]
    - The Deadlock Problem
    - Deadlock Prevention
    - Deadlock Detection and Recovery
    - Deadlock Avoidance
    - Graph Theoretic Models
    - Algorithmic Solutions to the Deadlock Problem

######### **Assignment #1 Due September 22th** #########

3. Multiprocessor Operating Systems [Sept 29, Oct 6] [Chapters 16, 17]
    - Multiprocessor System Architectures
    - Hardware Requirements and Setup for VM
    - Operating System Design Issues
    - Process Synchronization
    - Process Scheduling
    - Memory Management
    - Page Replacement Policies
    - Stack Algorithms

4. Distributed Systems [Oct 13 20] [Chapters 4, 5, 6]

######### **Assignment #2 Due Oct 13th** #########

    - Definition of DS
    - Problems of DS
    - Reliability
    - Resource Scheduling
    - Design Strategies
    - The ISO OSI Networking Model
    - Communication Primitives
    - Distributed Algorithms

- The Lamport Clock Algorithm
- Non-Token Based Distributed Mutual Exclusion Algorithms

5. EXAM 1, October 27, Chapters 2, 3,16, 17, 4, 5, and first half of 6

6. Token Based Distributed Mutual Exclusion Algorithms, Distributed Deadlock and Agreement Protocols [Nov 3, 10] [Chapters 6, 7, 8]
   - Deadlock Handling Strategies
   - Control Organizations
   - Distributed vs. Centralized Detection
   - Hierarchical Deadlock Detection
   - Classification of Agreement Problems
   - Applications of Agreement Algorithms

7. Distributed Resource Management [Nov 17, Dec 1] [Chapters 9, 10, 11]

######### **Nov 24 is Thanksgiving Day, No Class** #########

   - Distributed File Systems
   - Cache Consistency
   - Distributed Shared Memory
   - Memory Coherence and Associated Protocols
   - Distributed Scheduling
   - Load Distribution
   - Task Migration

######### **Assignment #3 Due Dec 1th** #########

8. EXAM 2, December 8, Chapters 6 (second half), 7, 8, 9, 10, 11
Last chance to submit any outstanding assigned work