Fall 2008

**CEG 434/634: Concurrent Software Design**

Yong Pei  
*Wright State University - Main Campus, yong.pei@wright.edu*

Follow this and additional works at: [https://corescholar.libraries.wright.edu/cecs_syllabi](https://corescholar.libraries.wright.edu/cecs_syllabi)

Part of the [Computer Engineering Commons](https://corescholar.libraries.wright.edu/cecs_syllabi), and the [Computer Sciences Commons](https://corescholar.libraries.wright.edu/cecs_syllabi)

**Repository Citation**

[https://corescholar.libraries.wright.edu/cecs_syllabi/92](https://corescholar.libraries.wright.edu/cecs_syllabi/92)

This Syllabus is brought to you for free and open access by the College of Engineering & Computer Science at CORE Scholar. It has been accepted for inclusion in Computer Science & Engineering Syllabi by an authorized administrator of CORE Scholar. For more information, please contact library-corescholar@wright.edu.
CEG 434/634
Concurrent Software Design

Syllabus

Fall Quarter, 2008

Time/Place: Lecture: 8:00 – 9:15 PM, Tu./Th., Millett 402

Instructor: Dr. Yong Pei, 489 Joshi Research Center
Tel. 937-775-5111, Email: yong.pei@wright.edu
Office Hours: 6:00-8:00 pm, Tu./Th.

GTA: No GTA.
Grader to be named soon.

Prerequisite: CS400, CEG433/633, Operating Systems.
Expected background: discrete mathematics, data structure, C or C++
programming experience in UNIX.

Course Description: This course provides an introduction to concurrent program design in
the UNIX environment. Classical problems of synchronization, concurrency, and their solutions are examined through course
projects and through readings on operating system design.

Text Books: Required: Operating System Concepts, 6th, 7th or 8th Ed., Silberschatz
and Galvin, Addison-Wesley, 2002.
Required: Unix Systems Programming: Communication,
Concurrency and Threads, 2nd Ed., Robbins and Robbins, Prentice
References: Interprocess Communications in Linux: The Nooks and

Website: CEG434-634 in WebCT.

Grading: Programming assignment – 30 %
Homework – 10 %
Midterm Exam – 30 %
Final – 30 %
Lectures:

The following tentative schedule defines in greater details what material is covered in the course and when it is covered.

<table>
<thead>
<tr>
<th>Week</th>
<th>Reading</th>
<th>Contents</th>
</tr>
</thead>
</table>
| 1    | Robbins Ch. 1  
      Silberschatz Ch. 1 | Welcome and introduction |
| 2    | Silberschatz Ch. 3, 5  
      Robbins Ch. 2 | Review: Process management, process scheduling, CPU Scheduling |
| 3    | Robbins Ch. 6, 8 | Basic UNIX inter-process communication  
                            Asynchronous events – UNIX signals |
| 4    | Robbins Ch. 18,20  
      Gray Ch. 10 | Client server computing  
                            Inter-process communication with socket |
| 5    | Gray Ch. 10 | Network Programming Using socket  
                            Midterm Exam (Thursday) |
| 6    | Silberschatz Ch. 4  
      Robbins Ch. 12 | Threads and POSIX threads |
| 7    | Silberschatz Ch. 6  
      Robbins Ch. 13,14 | Process synchronization (critical sections, semaphores, etc.) |
| 8    | Silberschatz Ch. 6  
      Robbins Ch. 13,14 | Case Study: Classic Synchronization Problems |
| 9    | Silberschatz Ch.7  
      Robbins Ch. 14 | Deadlocks |
| 10   | Silberschatz Ch.9 | * Selected Topics: Advanced Virtual Memory Management, middleware and network programming |
| 11   | | Final week |