Fall 2004

CEG 434/634: Concurrent Software Design

Thomas C. Hartrum
Wright State University - Main Campus, thomas.hartrum@wright.edu

Follow this and additional works at: https://corescholar.libraries.wright.edu/cecs_syllabi

Part of the Computer Engineering Commons, and the Computer Sciences Commons

Repository Citation
https://corescholar.libraries.wright.edu/cecs_syllabi/11

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, 2004

Time/Place: Lecture: 4:10 – 5:25 PM, M. & W., 154 Russ Engineering Center

Instructor: Dr. Thomas C. Hartrum, 258 Russ Engineering Center  
Tel. 937-775-5015, Email: thartrum@cs.wright.edu  
Office Hours: M, W, 2:00-3:00; T, Th 4:00 – 5:00.

GTA: Mr. Viraj Ambetkar, 326 Russ Engineering Center  
Email: vambetka@cs.wright.edu  
Office Hours: 3:00-4:00 pm, Monday & Wednesday.

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:  

Website: CEG434_634 in WebCT.  
www.cs.wright.edu/~thartrum

Grading: Programming assignment – 35 %  
Homework – 5%  
Midterm Exam – 25%  
Final – 35%
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>
<tbody>
<tr>
<td>1</td>
<td>Robbins Ch. 1</td>
<td>Welcome and introduction</td>
</tr>
<tr>
<td></td>
<td>Silberschatz Ch. 1</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Silberschatz Ch. 4</td>
<td>Process management, process scheduling,</td>
</tr>
<tr>
<td></td>
<td>Robbins Ch. 2, 3</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Silberschatz Ch. 6</td>
<td>CPU Scheduling</td>
</tr>
<tr>
<td>4</td>
<td>Robbins Ch. 6</td>
<td>UNIX I/O, inter-process communication</td>
</tr>
<tr>
<td>5</td>
<td>Robbins Ch. 8</td>
<td>Asynchronous events – UNIX signals, <strong>Midterm Exam</strong></td>
</tr>
<tr>
<td>6</td>
<td>Robbins Ch. 18,20</td>
<td>Client-server computing</td>
</tr>
<tr>
<td>7</td>
<td>Robbins Ch. 18,20 Gray Ch. 10</td>
<td>Inter-process communication with sockets</td>
</tr>
<tr>
<td>8</td>
<td>Silberschatz Ch. 7 Robbins Ch. 14</td>
<td>Process synchronization (critical sections, semaphores, etc.)</td>
</tr>
<tr>
<td>9</td>
<td>Silberschatz Ch. 5 Robbins Ch. 12, 13</td>
<td>Threads</td>
</tr>
<tr>
<td>10</td>
<td>Silberschatz Ch. 8</td>
<td>Deadlocks</td>
</tr>
<tr>
<td>11</td>
<td></td>
<td><strong>FINAL</strong> (Monday, 11/15/04, 5:45 PM – 7:45 PM)</td>
</tr>
</tbody>
</table>

Dept. of Computer Science & Engineering
Wright State University