CSC/CPE 569 Distributed Computing

1. CPE 569 Distributed Computing

2. credit units 4 contact hours 6

3. Course Coordinator: Chris Lupo


5. a. Course Description: Principles and practices in distributed computing: interprocess communications, group communications, client-server model, distributed objects, message queue system, distributed services, mobile agents, object space, Internet protocols. Distributed algorithms: consensus protocols, global state protocols. Fault tolerance: classification of faults, replication. 3 lectures, 1 laboratory.

   b. Prerequisite: CSC 348 and CSC/CPE 357; or graduate standing and consent of instructor.

c. Required/Elective/Selective Elective for CPE, CSC, EE, SE

<table>
<thead>
<tr>
<th>Required/Elective</th>
<th>CSC</th>
<th>CPE</th>
<th>SE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Required</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Elective</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Selective Elective</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

6. a. Course Goals/Outcomes

The student will be able to:
- Identify and critique the characteristics of distributed programs, distributed protocols, and distributed algorithms
- Classify distributed computing paradigms into a hierarchy
- Utilize a variety of application program interfaces (APIs) that support those the creation of distributed programs

b. How Student Outcomes addressed
(“B” = Basic level, “I” = Intermediate level, “A” = Advanced level)

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSC</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>I</td>
<td>I</td>
<td>A</td>
<td>N/A</td>
</tr>
<tr>
<td>SE/ CPE</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>B</td>
<td>I</td>
<td>A</td>
<td>I</td>
</tr>
</tbody>
</table>

7. Major Topics Covered: (number of lecture hours each)
• Introduction (3)
- Threaded Programming (3)
- Interprocess Communications (IPC) (3)
- Scalability (1)
- Map/Reduce (2)
- Scatter/Gather (1)
- Data Integrity (1)
- Compute Topologies (1)
- Client-Server Design Patterns (1)
- Group Communications (1)
- Distributed Objects (3)
- The Service-Oriented Architecture (1)
- Fault Tolerance (2)
- Distributed Algorithms (2)