Application for New Course

1. Submitted by the College of Engineering  
   Department/Division offering the course: Computer Science  
   Date: August 1, 2000

2. Proposed designation and Bulletin description of this course:
   (a) Prefix & Number: CS 670
   (b) Full Title: Distributed Operating System Theory  
       Abbreviated Title (≤ 24 characters): Distr Operating System
   (c) Lecture/Discussion hours per week: 3
   (d) Laboratory hours per week: 0
   (e) Studio hours per week: 0
   (f) Credits: 3
   (g) Course description:
       This course covers advanced distributed operating system algorithms and theory. Topics such as distributed mutual exclusion, distributed event ordering, distributed deadlock detection/avoidance, agreement protocols, consistent global snapshot collection, stable predicate detection, failure recovery, fault-tolerant consensus, leader election, process groups and group communication. Case studies of distributed operating systems such as LOCUS, Grapevine, V System, ISIS, Amoeba, Sprite, and Mach will be used as illustrations of the above algorithms.

   (h) Prerequisites (if any): CS 570 or consent of instructor.
   (i) May be repeated to a maximum of : Not applicable

4. To be cross listed as: Not applicable  
   Signature of cross-listing chair: ____________________________

5. Effective date:  
   Fall 2001

6. Course to be offered:  
   Fall  
   Spring  
   Summer  
   X

7. Will the course be offered annually; explain if not:

8. Why is the course needed? To provide a foundation in the concepts and design of Distributed Operating Systems.

9. (a) By whom will the course be taught?  
     D. Manivannan
   (b) Are facilities for teaching this course now available? Yes
       If not, what plans have been made for providing them?

10. What enrollment may reasonably be expected? 20
11. Will this course serve students in the Department primarily? Yes
   Will it be of service to a significant number students outside the Department? No
   If yes, under what area?

12. Check the category most applicable to this course:
   — traditional; offered in corresponding departments elsewhere;
   X relatively new, now being widely established;
   ___ not yet found in many (or any) other universities

13. Is this course part of a new proposed program? No
   If yes, which?

14. Will adding this course change the degree requirements in any programs? No
   If yes, explain:

15. Attach a list of the major teaching objectives of the proposed course and outline
   and/or a reference list to be used: See attached

16. If the course is a 100-200 level course, please submit evidence (e.g., correspondence)
   that the Community College System has been consulted.
   not applicable

17. Within the Department, who should be contacted for further information about the
   proposed course?
   Name: D. Manivannan or Raphael Finkel   Phone: 257-9234/257-3416

---

¹Approval of this course will constitute approval of the program change unless other program modifications are proposed.
Signatures of Approval:

Department Chair: ___________________________ Date: 11/14/2022
Dean of the College: ___________________________ Date: 2/8/21

Date of Notice to the Faculty: ___________________________ 1/21/01

Undergraduate Council: ___________________________ Date: ________
University Studies: ___________________________ Date: ________
Graduate Council: ___________________________ Date: 5/24/01
Senate Council: ___________________________ Date: ________

Date of Notice to the University Senate: ________

Action other than approval: ___________________________

If applicable, as provided by the Rules of the University Senate
**Needed Skills**
CS 670 is intended to be an advanced graduate-level course in distributed systems. Students should have had a graduate course in operating systems, (equivalent to CS 570) covering distributed operating systems, multiprocessor operating systems, database operating systems and security issues in distributed system.

**Learning Outcomes**
Students will learn about distributed systems design and implementation. They will be exposed to various areas of research in distributed systems and mobile computing systems. They will learn about designing and implementing fault tolerant distributed systems. A student completing this course successfully will be able to pursue independent research in distributed systems.

**Week by Week Course Outline**

<table>
<thead>
<tr>
<th>Week</th>
<th>Activities</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction</td>
</tr>
<tr>
<td>2-3</td>
<td>Synchronization, distributed mutual exclusion</td>
</tr>
<tr>
<td>4</td>
<td>Deadlock Detection/Avoidance</td>
</tr>
<tr>
<td>5</td>
<td>Consistent global snapshot collection</td>
</tr>
<tr>
<td>6</td>
<td>Predicate detection</td>
</tr>
<tr>
<td>7</td>
<td>Failure recovery in distributed systems</td>
</tr>
<tr>
<td>8-9</td>
<td>Fault-tolerant consensus</td>
</tr>
<tr>
<td>10-11</td>
<td>Leader election algorithms, Agreement Protocols</td>
</tr>
<tr>
<td>12-13</td>
<td>Process Groups and group communication</td>
</tr>
<tr>
<td>14-15</td>
<td>Experimental distributed operating systems</td>
</tr>
</tbody>
</table>

**Examinations**
Exact details about examinations in this course will be determined by the instructor offering the course. Typically there will be one in-class, midterm examination during the semester and a two-hour final examination. Specific details will be made available in the syllabus at the start of each semester in which the course is offered.

**Grading**
A student's grade will be determined by a weighted average of homework assignments, programming exercises, projects, hour examinations, and the final examination. The faculty offering the course will make the details available at the start of the course. A typical weighting is:

- Homework: 40%
- Midterm: 25%
- Final Examination: 35%