COMP 4270/6270: Introduction to Operating Systems – Spring 2008
Instructor: Prof. Santosh Kumar
5:30pm –6:55pm, Tuesday, Thursday in FIT 227

Office: 376 Dunn Hall	**Department Office:** 209 Dunn Hall
Phone: (901) 678 2487	**Department Phone:** (901) 678-5465
E-mail: santosh.kumar@memphis.edu	**URL:** http://www.cs.memphis.edu/~santosh
TA: Justus Nyagwencha (juyamweya@hotmail.com)	

Office Hours:

Monday	Tuesday	Wednesday	Thursday	Friday
9 – 11 (GA)	4 – 5 (Instructor)		4 – 5 (Instructor)	9 – 11 (GA)
Also by Appointment				

Course Description:

COMP 4270/6270. Operating system services, operating system structures; process and thread scheduling; management of asynchronous processes – concurrency, process synchronization, semaphores, monitors; deadlocks; memory management – paging, segmentation, and virtual memory; file system – access and allocation; hierarchy of storage devices. PREREQUISITE: COMP 3160, and either COMP 3410 or EECE 4278.

Why take this course?

1. To have a basic understanding of an operating system;
2. To uncover the mystery of modern operating systems;
3. To learn programming for smart dust (wireless sensors).

Resources:

Required Text

- Operating System Concepts, 7th Edition, Silberschatz, Galvin, and Gagne, Wiley, 2005.

Recommended Texts

- The Design of the UNIX Operating System, Maurice J. Bach, Prentice Hall, 1986.
- Operating Systems: Internals and Design Principles, William Stallings, 5th Edition, 2005.

Other Resources:

- TinyOS Tutorial: http://www.tinyos.net/tinyos-1.x/doc/tutorial/
- TinyOS Webpage: http://www.tinyos.net/
- TinyOS Programming by Phil Levis (online book)

Intended Outcomes:

- Theoretical Concepts:
    1. The student will be able to identify basic roles of an operating system(OS).
    2. The student will be able to list the various components of an OS.
    3. The student will be able to identify major issues in the design of an OS.
    4. The student will be able to analyze the tradeoffs involved in the design of an OS.
    5. The student will be able to describe recall well-known algorithms used in operating system designs.
    6. The student will be able to modify the TinyOS source code to test alternative algorithms used in various OS components.
- Programming Skills – *Wireless Sensor Network Platform (Mote)*:
    1. The student will be able to write simple NesC programs for the mote platform.
    2. The student will be able to write and test simple NesC programs for communication between mote units.

Class Format:

The class will involve lectures by the instructor, individual and team-based homework, paper presentations by students, and a team-based project. Basic concepts for each topic will first be introduced by the instructor in lectures. Homework will then be assigned to enhance the understanding of basic concepts.

Prior Class Preparation:

Thorough preparation—by students *and* instructor—and active participation are essential to a successful course. Learning comes from struggling with the issues outside of class, then discussing them (and the struggle) in class. Unprepared students personally miss out on most of the learning and also cheat their classmates because they cannot contribute fully to the learning that occurs in class.

The instructor will assign readings from books and papers. Each student is expected to have read these before coming to class. This will enhance student learning as well as enhance other students' learning because more meaningful discussion can take place in the class. Class participation assessment will be done by the instructor as well as peer students in the class.

Assignments, Exams, and Project:

Exams: There will be two midterm exams – one midway through the semester and one in the last class. The exams will be open book and open notes but closed neighbor and closed computers (laptops, desktops, PDAs, cell phones, etc.). The 2nd midterm exam will not be comprehensive.

Homeworks: There will be both individual homeworks and team homeworks based on the course content.

Presentations: There will be two 15-minute team presentations. The presentations will be evaluated by the entire class.

Project: Each student team will do a programming project on the wireless sensor network mote platform. See the TinyOS webpages listed in the "Other Resources" Section for tutorials on how to program the mote platform.

A list of options for the project will be provided. However, students are free and encouraged to propose their own projects. No two teams will work on the same project.

A series of programming assignments will prepare the students for their projects. Upon successful completion, the projects will be demonstrated to the public in the final exam time slot. Each project will be accompanied with a short presentation.

You are encouraged to discuss assignments with the instructor and/or the GA for feedback and hints.

Teams:

Each student is expected to form/join a team for both homework and the project. The team composition for homework may be different than for the project. Team membership for homework will frequently be rotated to allow students to know and learn from different students in the class.

During the first week of class, students will form teams of 2 unless the instructor deems a different team size is warranted. Formation of teams will be left to your discretion, but I encourage you to include some variety in terms of gender, ethnicity, nationality, work experience, etc. If you need motivation beyond the opportunity to learn from classmates with different experiences, recognize that the teams you work with on the job usually include such diversity. It is wise to have at least one team member who is a fluent in English, to help ensure that your reports are written clearly.

Teams are to work *independently*. Reports, programs, or solutions from students who took the class in the past are strictly off limits.

Evaluation:

Final Grades:

An individual's grade will be composed of his/her team's score as well as his/her individual score as described in the following table.

Team Evaluation		***Individual Evaluation***	
Homework	15%	Homework	15%
Presentations (2)	10%	Midterm Exam 1	15%
Project	15%	Midterm Exam 2	15%
Programming Assignments	10%	Class Participation	5%

Assignment of letter grade will be determined based on performance of the class. Current plan is:

A+: >= 95, A: >= 90, A-: >= 87.5, B+: >= 82.5, B: >= 80, B-: >= 77.5, C+: >= 72.5, C: >= 70, C-: >= 67.5, D+: >= 62.5, D: >= 60, F: < 60.

Course Policies:

Attendance:

You are required to attend every class *unless there is a documented emergency*. The instructor may check attendance at the beginning of every class. If you miss a class, you will have to make your own arrangements to learn the materials covered in that class and to know of any announcements made in that class.

Late Policy:

Homework and reports are due *before class on the due date*. For every 24 hours that an assignment is late, 20% of the total score will be deducted. For every day that an assignment is late, 20% of the total maximum credit will be deducted. For example, if an assignment is worth a maximum of 10 points, it will be worth only a maximum of 8 points if the assignment is late by one day.

Any homework or reports submitted 5 days after the due date and time will **NOT** be accepted (submit all your homework on eLearn).

Testing Policy:

There will **NOT** be any makeup quizzes or exams *unless there is a documented emergency*, so it is very important for you to attend every lecture and exam.

Plagiarism/Cheating Policy: (These paragraphs are mandatory.)

Plagiarism or cheating behavior in any form is unethical and detrimental to proper education and ***will not be tolerated.*** All work submitted by a student (projects, programming assignments, lab assignments, quizzes, tests, etc.) is expected to be a student's own work. The plagiarism is incurred when any part of anybody else's work is passed as your own (no proper credit is listed to the sources in your own work) so the reader is led to believe it is therefore your own effort. Students are allowed and encouraged to discuss with each other and look up resources in the literature (including the internet) on their assignments, but ***appropriate references must be included for the materials consulted***, and appropriate citations made when the material is taken verbatim.

If plagiarism or cheating occurs, the student will receive a failing grade on the assignment and (at the instructor’s discretion) a failing grade in the course. The course instructor may also decide to forward the incident to the University Judicial Affairs Office for further disciplinary action. For further information on U of M code of student conduct and academic discipline procedures, please refer to: **http://www.people.memphis.edu/~jaffairs/**

Course Outline
List lecture topics or chapter sections by week or lecture meeting days.

Lecture	Lecture Topics (Tentative)
1	Course Overview and Introduction to OS (Ch 1)
2	Introduction to OS (Ch 1)
3	Introduction to TinyOS and Demonstration (Somnath Mitra)
4	Operating System Structures (Ch 2) (GA)
5	Operating System Structures (Ch 2) contd.
6	Processes (Ch 3)
7	Processes (Ch 3) contd.
8	Threads (Ch 4)
9	CPU Scheduling (Ch 5)
10	CPU Scheduling (Ch 5) contd.
11	Process Synchronization – Semaphores, Monitors, Atomic Transactions (Ch 6)
12	Students Presentations
13	Student Presentations
14	Process Synchronization contd.; Review
15	Midterm Exam 1
16	Deadlocks – Avoidance, Detection, and Prevention (Ch 7)
17	Deadlocks contd.
18	Memory Management (Main Memory) (Ch 8)
19	Memory Management (Main Memory) (Ch 8)
20	Memory Management (Virtual Memory) (Ch 9)
21	Memory Management (Virtual Memory) (Ch 9)
22	File System Interface (Ch 10)
23	File System Implementation (Ch 11)
24	Mass Storage Structure (Ch 12)
25	Student Survey Presentations (Novel OS technologies)
26	Student Presentations contd.; Review
27	Midterm Exam 2
Finals Week	Project Demonstration and Presentation