CMSC 412: Operating Systems

Neil Spring

Fall 2008

Instructor Neil Spring

E-mail nspring@cs (include "412" in your subject line)

Class TuTh 3:30-4:45 CSIC 1122

Office hours TBA AVW 4133

TA Aaron Schulman: schulman@cs (include "412" in your subject line)

Web `http://www.cs.umd.edu/class/fall2008/cmsc412/`

Textbook Silbershatz, Galvin & Gagne, *Operating System Concepts*, 8th ed.

1 Goals of this course

At the end of this class, you should be able to write a device driver, modify an operating system, understand how operating systems help or interfere with applications, write concurrent programs without deadlock, and know what's missing when you program for little devices.

Few of these goals are directly addressed by the course exercises (we likely won't be modifying linux or writing device drivers) but these are the goals I plan to use to decide what material to cover.

2 Summary

The course will cover the following core topics:

Processes What makes a process, how are they run concurrently, how to create them and communicate between them.

Threads What makes a thread, libraries.

Scheduling How to keep interactive applications responsive and background applications make forward progress.

Synchronization and Deadlock Locks on shared data, and preventing cooperative processes from getting stuck.

Memory and Virtual Memory Swapping, paging, segmentation, allocating memory, copy-on-write, etc.

File System Interface and Implementation the function calls, mounting file systems, organizing blocks on disk, allocation, recovery.

Disk and Storage Systems disk scheduling, RAID, tape hierarchies.

I/O Systems programmed and interrupt-driven I/O.

And given time, the following additional topics:

Protection capabilities, defining access control.

Distributed Coordination Events, atomicity, deadlock in distributed systems where messages can be lost.

Linux how each of the features we learned about are implemented in Linux.

Security Basic crypto, authentication.

Distributed Systems Distributed communication primitives.

Distributed File Systems Global naming of files.

iPhone Application Sandbox Depending on the state of Apple's NDA, there's a different model in securing applications for the same user written by developers of limited trust on a device with lots of personal information. (not in the textbook, obviously.)

This structure is intended to follow the textbook. I will make exceptions to the order to support the programming assignments as needed.

3 Prerequisites

Experience in CMSC417 (networks) may help you.
CMSC311 or ENEE350 – Computer Organization.
CMSC330 – Programming Languages.

You must know what a function pointer is and how it is used. Find a book on C today if you do not.

You should understand basic issues of concurrency. That includes the interactions between non-blocking sockets, user-level and kernel-level threads, locking, etc. Too many students seem to think that forking a thread will solve a simple problem without creating many more.

4 Style

I don't use lecture slides. I expect to be interupted. I will assume you know more than you do; it is your job to pay attention, and make me clarify when I've left you behind.

5 Grading

5.1 Participation: 5%

In a class so large, I can't expect each of you to speak; participation here is a negative grade, if I think you're doing poorly and it's your own fault for not being engaged with the material, you won't get the participation bump.

5.2 Homeworks and Cell-Phone Quizzes: 10%

A few assignments to take home and complete will be graded. Quizzes given when cell phones ring are included here.

Quizzes given in class when cell phones ring will also be graded here. You will dread the cell phone quiz if you allow yourself to fall behind.

5.3 Midterm Exam: 20%

5.4 Final Exam: 30%

The midterm and final exams will mix multiple choice, simple matching, short answer and long answer questions. The midterm will consume a lecture slot, the final during finals week as scheduled by the university. The exams will be longer than will allow all of you to finish the entire exam. You will have to learn and study before the exam.

5.5 Programming Assignments: 35%

The programming assignments in this class will use GeekOS. The TA and I are working to permit running the assignment within the QEMU emulator, which has the advantage of being available for the mac. The assignments are difficult.

6 Lateness

All programming assignments can be turned in electronically. I will permit one programming assignment to be turned in after the weekend (when due Friday, it can be turned in on Monday). I expect any data loss due to dogs, roommates, lightning strikes or FBI confiscating your machine can be dealt with over a weekend.

7 Administrative Cruft

I dislike this section greatly, but codifying each of these policies is important for keeping myself sane and making clear what my expectations are. I'd much prefer a section that said "treat me with respect and I'll do the same for you;" this section is intended mostly for those who would hope to game the system. Note that I copied verbatim some of these passages; I hope you appreciate irony.

7.1 Excused absences

Students claiming a excused absence must apply in writing and furnish documentary support (such as from a health care professional who treated the student) for any assertion that the absence qualifies as an excused absence. The support should explicitly indicate the dates or times the student was incapacitated due to illness. Self-documentation of illness is not itself sufficient support to excuse the absence. An instructor is not under obligation to offer a substitute assignment or to give a student a make-up assessment unless the failure to perform was due to an excused absence. An excused absence for an individual typically does not translate into an extension for team deliverables on a project.

7.2 Religious observances

I have made an effort to avoid deadlines 9/29-10/1 and 10/8-10/9. Please inform me in advance of religious observances that will interfere with your ability to complete assignments on time.

7.3 Honor code

The University of Maryland, College Park has a nationally recognized Code of Academic Integrity, administered by the Student Honor Council. This Code sets standards for academic integrity at Maryland for all undergraduate and graduate students. As a student you are responsible for upholding these standards for this course. It is very important for you to be aware of the consequences of cheating, fabrication, facilitation, and plagiarism. For more information on the Code of Academic Integrity or the Student Honor Council, please visit `http://www.studenthonorcouncil.umd.edu/whatis.html`.