

Prof: Dr. Peter M. Maurer
Office: Rogers 220.08
Phone: 710-7305
Email: Peter_Maurer@Baylor.edu

Text: Silberschatz, Galvin, Gagne
Operating System Concepts
Seventh Edition
Office Hours: MWF 8-10AM, 11-12AM
TR 8-9:30AM, 11-12AM

Blog: <http://csi4337.petermmaurer.com>

All class members are required to subscribe

1. Jan 9 Syllabus and Class Roll.
2. Jan 11 What and why is an operating system?
3. Jan 16 Hardware Organization
4. Jan 18 Operating System Structure
5. Jan 23 Processes and Threads
6. Jan 25
7. Jan 30 CPU Scheduling
8. Feb 1
9. Feb 6 Exam Review
10. Feb 8 Exam #1
11. Feb 13 Process Synchronization
12. Feb 15
13. Feb 20 Deadlocks
14. Feb 22
15. Feb 27 Memory Management
16. Mar 1 Virtual Memory
17. Mar 6
18. Mar 8
19. Mar 20 Review for exam
20. Mar 22 Exam #2
21. Mar 27 File Systems (in memory)
22. Mar 29 File Systems (on disk)
23. Apr 3
24. Apr 5
25. Apr 10 I/O Systems and Mass-Storage Structure
26. Apr 12 Distributed System Structures
27. Apr 17 Distributed File Systems
28. Apr 19 Diadeloso
29. Apr 24 Distributed Coordination
30. Apr 26 Review for Final
- Final Exam: Thursday May 4, 2:00-4:00 PM (Rogers 104)

Course Objectives

By the time you have finished with this course, you should be familiar enough with the principles of operating systems that you could (at least theoretically) construct your own operating system for a new computer. To accomplish this goal, there are several smaller objectives that we must meet. These are as follows.

1. Learn where the boundary lies between hardware and software. When you interact with any computer system, part of the interaction is handled by the hardware (moving mouse-ball, depression of keyboard keys) and part of the interaction is handled by the software (displaying characters on the screen). We need to know where the boundary lies so that we know what is available in the hardware, and what must be implemented in software.
2. Learn how operating system code gets executed. Despite the sophistication of today's computers, for the most part they still execute only one instruction at a time. That means when a user program is executing, the operating system must be idle. One of the most important things you will learn is the mechanisms that are used to give the operating system its "turn to execute."
3. Learn the types of tasks that are normally relegated to an operating system. Some things that seem to be part of the operating system are not. For example the LINUX shell is not part of the LINUX operating system.
4. Learn the most common and popular algorithms and data structures for performing standard operating system tasks. If an operating system must perform a particular task for virtually every program (memory allocation is an example) then you should know the most common methods for performing the task.

Grading

Final Exam: 35%

Projects and homework: 15%

Other Exams: 50% -- Equally divided among all exams.

Other Information

University attendance policy will be enforced.

The class blog is located at URL <http://csi4337.petermmaurer.com> All information regarding this class will be distributed through this blog. If I post something on this blog I will expect that you have seen it. You can subscribe through E-Mail or through RSS. Make sure if you use E-Mail that it is an address you check at least once a day. If you use an RSS feed, make sure your reader is running all the time and is uncluttered enough that you won't miss anything.

Do not leave early!

Do not come late!

I have an open door policy with respect to students. I'm in my office most of the time. I am willing to meet with you any time I am in my office. Feel free to come to me with any matter that is troubling you, even if it has nothing to do with the class.