Prof: Dr. Peter M. Maurer
Office: Rogers 220S
Phone: 710-7305
Email: Peter_Maurer@Baylor.edu
Web: http://cs.ecs.baylor.edu/~maurer

Text: Silberschatz, Galvin, Gagne
Operating System Concepts
Sixth Edition

Office Hours: 8-11 MWF, 8-9:30,11-12 TR

1. Jan 13	Introductions	
2. Jan 15	What and why is an operating system?	Chapter 1
3. Jan 20	Hardware Organization	Chapter 2
4. Jan 22	Operating System Structure	Chapter 3
5. Jan 27	Processes and Threads	Chapters 4 and 5
6. Jan 29		
7. Feb 3	CPU Scheduling	Chapter 6
8. Feb 5		
9. Feb 10	Review	
10. Feb 12	Exam #1	
11. Feb 17	Process Synchronization	Chapter 7
12. Feb 19		
13. Feb 24		
14. Feb 26	Deadlocks	Chapter 8
15. Mar 2		
16. Mar 4	Memory Management	Chapter 9
17. Mar 9		
18. Mar 11		
19. Mar 23	Review	
20. Mar 25	Exam #2	
21. Mar 30	Virtual Memory	Chapter 10
22. Apr 1		
23. Apr 6		
24. Apr 8	File Systems	Chapter 11
25. Apr 13		
26. Apr 15		
27. Apr 20	Distributed Computing	Chapters 15-17
28. Apr 22		
29. Apr 27	Linux	Chapter 20
30. Apr 29	Review	
Final Exam:	Monday May 10, 9:00-11:00 AM	

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: 15%
Other Exams: 50% -- Equally divided among all exams.

Other Information

Exam grades will be curved, if necessary – but it probably won't be necessary.

University attendance policy will be enforced.

You are expected to attend every class. If you are unable to attend a particular class, you are still responsible for the material covered in the class. You must make arrangements to obtain this material from another student. Lectures will not be repeated.

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.