Prof: Dr. Peter M. Maurer
Office: ECS 305F
Phone: 710-7305
Email: Peter_Maurer@Baylor.edu
Web: TBA

Text: Silberschatz, Galvin, Gagne
Operating System Concepts
Sixth Edition
Office Hours: 9-12 M-F

1. Aug 26	Introductions	
2. Aug 28	What and why is an operating system?	Chapter 1
3. Aug 30		
4. Sep 2	Hardware Organization	Chapter 2
5. Sep 4		
6. Sep 6		
7. Sep 9	Operating System Structure	Chapter 3
8. Sep 11		
9. Sep 13	Processes and Threads	Chapters 4 and 5
10. Sep 16		
11. Sep 18		
12. Sep 20	CPU Scheduling	Chapter 6
13. Sep 23		
14. Sep 25		
15. Sep 27	Exam Review	
16. Sep 30	Exam #1	
17. Oct 2	Process Synchronization	Chapter 7
18. Oct 4		
19. Oct 7		
20. Oct 9	Deadlocks	Chapter 8
21. Oct 11		
22. Oct 14	Memory Management	Chapter 9
23. Oct 16		
24. Oct 18	Fall Break	
25. Oct 21	Virtual Memory	Chapter 10
26. Oct 23		
27. Oct 25		
28. Oct 28	File Systems (in memory)	Chapter 11
29. Oct 30		
30. Nov 1	Review for Exam #2	
31. Nov 4	Exam #2	
32. Nov 6	File Systems (on disk)	Chapter 12
33. Nov 8		
34. Nov 11	I/O Systems and Mass-Storage Structure	Chapters 13 and 14
35. Nov 13		
36. Nov 15		
37. Nov 18	Distributed System Structures	Chapter 15
38. Nov 20		
39. Nov 22		
40. Nov 25	Distributed File Systems	
41. Nov 25		
42. Nov 27	Thanksgiving	
43. Nov 29	Thanksgiving	
44. Dec 2	Distributed Coordination	
45. Dec 4		
46. Dec 6	Exam ?	
47. Dec 9	Review for Final	
Final Exam:	Friday Dec 13, 1:30-3:30 PM	

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: 25%
Projects: 15%
Other Exams: 60% -- 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.