

OPERATING SYSTEMS
COP 6611
SPRING 2012
TUE/THU 11:00AM-12:15PM, ENC 1002

Instructor:	Anda Iamnitchi	Office:	ENB 334
Email:	anda@cse.usf.edu	Phone:	974-5357
Office Hrs:	Tue/Thu 4-5pm	Fax:	974-5456

TAs:	Matthew Shreve	Amin Ahmadi Adl
Email:	mshreve@mail.usf.edu	amin1@mail.usf.edu
Office Hrs:	-	Tue/Thu 5-6:30pm
Office:	-	ENB 335

Course website: <http://www.cse.usf.edu/~anda/cop6611>

COURSE OVERVIEW

This course covers an exciting range of materials from the broad field of operating systems, including basic operating system structure, memory management, file systems and storage, distributed systems, virtual machines, and security. We will examine influential historical systems and important current efforts, extracting lessons both on how to build systems as well as how to evaluate them.

The textbook material will be supplemented with research papers that are made available on the class web page.

PREREQUISITES

- Undergraduate Operating Systems

TEXTBOOK (REQUIRED)

Operating Systems Concepts, Silberschatz, Galvin & Gagne, 8th Edition.

CLASS STYLE

Much of the textbook material is basic undergraduate operating systems material. Some of you have that information still fresh in your minds, while others less so. A traditional lecturing style from the textbook material will either frustrate those of you comfortable with the undergraduate material or be too fast paced for those of you who need a significant refresher.

Consequently, the class format this year will be highly interactive, with most of the class time dedicated to question answering, paper discussions, and problem solving. This departure from the traditional lecturing style is based both on research in education (the numbers will surprise you!) and on experience with and feedback from previous classes.

You will be required and expected to be comfortable with the assigned chapters from the textbook on your own (i.e., read carefully or skim, depending on your level). I will emphasize the important concepts you have to look for while reading and we will discuss the challenging aspects in class. Class time will be dedicated to short quizzes on the assigned textbook reading for the class (10-15 minutes); in-group or individual problem solving sessions (10-15 minutes); your questions and discussions of research papers (45 minutes).

WORKLOAD AND GRADING

TEXTBOOK MATERIAL (HONOR POINTS): for each chapter submit about one page of notes summarizing the most important concepts. Write the notes as for your own use (e.g., to revisit later before the qualifiers or final exam). Chapter notes are due at midnight before the class when we will start discussing the material in the chapter, but you are welcome to improve the notes during the semester. This component will not be formally graded; the reading notes will be used for deciding borderline situations (e.g., good notes showing understanding of the material may bring a grade from 88% to A). Textbook notes may accumulate to up to 5% of the grade; not submitting the notes will not have grade penalties.

PAPER SUMMARIES, READING QUESTIONS, IN-CLASS PARTICIPATION (25%): In addition to the material covered in the textbook, we will read and discuss about 13 research papers. Some of these papers form the basis of the current state-of-the-art in operating system, while others are hot topics or current trends. These papers will provide both depth and breadth in the subject and will be a gentle introduction in systems research.

You are expected to submit concise paper summaries (less or about 1/2 page) and precise answers to questions on the paper by midnight before the Thursday of the corresponding week. The summaries and answers are to be submitted via H2O (an Internet project which lets you submit ideas and conduct discussions with ease – you will be asked to join as a member). Each paper will have a deadline by which the summaries and answers have to be posted. You will be able to see your colleagues' answers after the deadline. The questions for each paper will be posted at least one week before the due date. The summaries should touch the following and should provide you with nice notes for when studying for exams:

- **Summary:** 1-2 sentences that give the essence of the paper. Please focus on the solution proposed rather than only on the problem. For example, “This paper describes the design of an operating system” should rather be “This paper describes a multiprogramming operating system with a layered design, where layers are responsible for disjoint sets of functionalities”.
- **Problem:** a description of the problem the paper attempts to solve. Make sure you identify the assumptions stated in the problem description.
- **Contributions:** the main ideas of the solution proposed.
- **Discussion:** any points you may want to raise, or questions you might want to ask the authors or see discussed in class.

Participation in class discussions is rewarded by at least 5% of the credit allocated for this component of the course. It is quality that matters more than quantity, so try to think of questions that would lead to interesting discussions on the relevant material, present and discuss less obvious aspects of the paper, find meaningful parallels with other relevant/related systems, etc.

PROGRAMMING ASSIGNMENTS (25%): There will be 4 small programming assignments related to the textbook material:

1. Working with Pthreads on Linux. [due end of 3rd week] (5%)
2. A synchronization problem. [due end of 6th week] (5%)
3. File system performance measurement. [due end of 9th week] (10%)
4. Socket programming. [due end of 15th week] (5%)

All assignments are individual. All assignments should be submitted via the blackboard in a zip file. The code has to be documented. A readme file containing instructions on executing the program should also be provided.

HOMEWORK ASSIGNMENTS (20%): 4-6 problem sets will be assigned. Assignments are due before the beginning of the class and should be submitted via Blackboard (thus, typed in).

EXAMS (30%): Two in-class exams will make the rest of the grade, 10% for the midterm and 20% for the final. The exams will test understanding of the class material (covered in textbook, reading list, and class lectures). Samples of exam questions will be provided early in the semester. The final exam is comprehensive.

GRADING SCHEME AND EXTRA CREDIT: There will not be any extra credit or make up work. I will most likely not use grade curving.

- A: 90-100%
- B: 80-90%
- C: 70-80%
- D: 60-70%
- F: below 60%

POLICIES

ATTENDANCE: Class meetings will be interactive and you are expected to participate in a meaningful way. In-class work will be assigned during each session for you to solve it in a group or alone.

Students must provide written notice to the instructor at the beginning of each academic term if they expect to be absent for a class or announced examination for the observance of religious holy days.

MISSED EXAMS: If you miss the midterm exam, the final exam will replace it.

DEADLINES AND LATE WORK: Answers to reading questions must be submitted by midnight before the Thursday class in the weeks they are due. No late submissions will be

accepted. Problem assignments are at the beginning of class on the day they are due. All programming assignments deadlines are 11:59pm on the day they are due. Late assignment submission is not accepted (will result in a 0).

COLLABORATION AND CHEATING: Collaboration is strongly encouraged on reading. You may want to form a reading group to better understand the research papers. In this case, you still have to submit your own paper summaries and answers; in addition, please acknowledge the colleagues in your reading group.

Collaboration on all class components must be acknowledged. If you use any published or unpublished work in your own work, you must give full citation.

Acts of cheating and plagiarism will be reported to the appropriate administrative bodies. You are responsible for knowing and will be held to the Honor Code of the University.

This course requires you to submit your work to a plagiarism detection site that will be identified by your instructor. In order to comply with federal (FERPA) and state privacy laws, you (students) are not required to include personal identifying information such as your name, SSN, and/or U# in the body of the work or use such information in the file naming convention prior to submitting. Please follow carefully the instructions regarding what identifying information to include. Your submission will be placed in the course grade center in your account and thus can be attributed to you.

ACCOMMODATIONS: Students in need of academic accommodations for a disability may consult with the office of Students with Disabilities Services to arrange appropriate accommodations. Students are required to give reasonable notice prior to requesting an accommodation.

CLASS RECORDINGS: Please do not sell notes from or record class lectures without my permission.

LECTURE SCHEDULE

Week	Topics	Reading
1, 2	Overview. OS Structures	Ch. 1, 2, 23, [1,2,3]
3, 4	Process Management. Threads. Scheduling	Ch. 3, 4, 5, [4]
5, 6	Process Synchronization. Deadlocks	Ch. 6, 7 [5]
7	Memory Management	Ch. 8, 9, [6]
8	File Systems	Ch. 10, 11, [7]
9	Distributed File Systems. Midterm exam.	Ch. 16, 17, [8]
10	Distributed Storage	Ch. 12, [9]
11	Protection and Security	Ch. 14, 15
12, 13	Distributed Systems. Distributed Coordination	Ch. 18, [10]
14	Hot Topics in OS (1) – TBD	[11, 12]
15	Hot Topics in OS (2) – TBD	[13]

Reading List:

1. D. Engler, F. Kaashoek and J. O'Toole, "Exokernel: An Operating System Architecture for Application-Level Resource Management ", Proceedings of the 15th ACM Symposium on Operating System Principles, ACM, December 1995.
2. J. Liedtke, " On Micro-Kernel Construction ", Proceedings of the 15th ACM Symposium on Operating System Principles, ACM, December 1995.
3. P. Barham, B. Dragovic, et al. Xen and the Art of Virtualization. In the Proceedings of the ACM Symposium on Operating Systems Principles (SOSP), 2003
4. C. Waldspurger and W. Wehl. Lottery Scheduling: Flexible Proportional-Share Resource Management in Proceedings of 1st USENIX Symposium on Operating System Design and Implementation, 1994.
5. J. M. Mellor-Crummey and M. Scott. Algorithms for Scalable Synchronization on Shared-Memory Multiprocessors, *ACM Transactions on Computer Systems*, Feb. 1991.
6. C. Waldspurger. Memory Resource Management in VMware ESX Server in Proceedings of the 5th Symposium on Operating Systems Design and Implementation, 2002.
7. M. Rosenblum and J. K. Ousterhout. The Design and Implementation of a Log-Structured File System. *ACM Trans. on Computer Systems* 10(1), 1992, pp. 26-52.
8. S. Ghemawat, H. Gobioff and S-T Leung. The Google File System, in Proceedings of the 19th ACM Symposium on Operating Systems Principles, 2003.
9. D. A. Patterson, G. Gibson, and R. H. Katz. A Case for Redundant Arrays of Inexpensive Disks (RAID) Proceedings of the 1988 ACM SIGMOD Conference on Management of
10. L. Lamport, Time, Clocks, and the Ordering of Events in a Distributed System, *Communications of the ACM*, July 1978, pages 558-564.
11. Corey: An Operating System for Many Cores. Silas Boyd-Wickizer, Haibo Chen, Rong Chen, Yandong Mao, Frans Kaashoek, Robert Morris, Aleksey Pesterev, Lex Stein, Ming Wu, Yuehua Dai, Yang Zhang, Zheng Zhang, OSDI 2008.
12. TBD
13. TBD

Every part of this syllabus is subject to adjustment as the semester progresses. Please contact me as soon as possible if you have particular interests in material that is relevant to the class topic but not covered in enough detail; I will be happy to accommodate reasonable requests for modifications.