General Information

COP 4600 - Operating Systems. Fall 2009: Tu, Th 4:30-5:45 PM, HEC 117

Instructor: Dan C. Marinescu
Office Hours: TU, Th 3-4 PM. Office Location: HEC 439B.
TA: Chen Yu. Email: cyu@knights.ucf.edu. TA's office: HEC 354.
TA's office hours: M-W 1:00 - 3:00. PM
Final Exam: Thursday December 10, 2009, 4:00 - 6:50. PM

Course Objectives: We have revised the COT 4600 class for the Fall 2009 semester to give the students a fresh look at basic principles guiding the design and implementation of operating systems. The focus of the class is switched from the discussion on "how" operating systems are implemented to the identification of the most important questions the designer of an operating system has to address and "why" a solution is better than others. Another major departure from the more traditional approach in covering operating systems is the emphasize on performance; several lectures cover computer system performance analysis. We also emphasize the "big picture" the relationship of operating systems with other subjects from undergraduate curriculum including: computer architecture/computer organization, programming languages, algorithms, networking, databases, modelling and performance analysis. This course covers topics on the engineering of computer software and hardware systems: techniques for controlling complexity; strong modularity using client-server design, virtual memory, and threads; networks; atomicity and coordination; performance. The discussion is anchored in reality and abstract subjects are followed by an in depth analysis of case studies.

Prerequisites: A computer architecture class; Programming language (C, C++, Java), or consent of the instructor.

Textbook: "Principles of Computer Systems Design; An Introduction" by Jerome Saltzer and Frans Kaasohoek. Publisher: Morgan Kaufmann, ISBN 978-0-12-374957-4. The new textbook for this class was published in July 2009 and reflects the experience of the two authors in teaching the subject at MIT for many years. One of the authors of the textbook, Prof. Jerome Saltzer, helped formulate the undergraduate curriculum in Computer Science, and developed the core subject on computer systems engineering at MIT; In mid 1960s he was involved in all aspects of the design and implementation of the Multics system and more recently in the design of the Kerberos authentication system. The topics covered by the six chapters of the book are:

1. Systems
2. Elements of Computer System Organization
3. The Design of Naming Schemes
4. Enforcing Modularity with Clients and Services
5. Enforcing Modularity with Virtalization
6. Performance

Additional references: Suggestions for the most relevant readings on the topics covered in the text are are given on pages 375-423.

Assignments: There are several homework assignments and a class project. The class project is to simulate the operation of a simple kernel for a computer system. It involves multiple phases:

1. Simulate a processor with a minimal instruction set operating in kernel and user mode. Due week 4.
2. Virtualize the memory. Design and implement a paging system and a virtual memory manager. Due week 8.
3. Virtualize the processor. Add a thread management system. Due week 10.
4. Add a virtual communication channel allowing threads to communicate using a bounded buffer and *send* and *receive* primitives. Due week 14.

Class attendance and grading: The students are strongly encouraged to attend every class and be active. Our policy is simple, based on professional standards: on exams you should not collaborate. On all other assignments you are welcome to work with anyone else on ideas and understanding, but your writing should be your own and you should carefully acknowledge all contributions of ideas by others, whether from classmates or from papers you have read. Acts of academic dishonesty will not be tolerated; when detected they lead to unconditional failure of the class. The grading scheme is :

- Homework: 15%
- Project: 35%
- Midterm: 20%
- Final: 30%

Communication with the students: The students are encouraged to come to the office hours of the instructor to discuss topics covered during the lectures and to the office hours of the TA to discuss the assignments. If a student has a question regarding grading of the assignment s(he) should first discuss it with the TA and only if the issue is not resolved bring it to the attention of the instructor. Information regarding the class, including the assignments will be posted on the class site: http://www.cs.ucf.edu/ dcm/Teaching/OperatingSystems4600/Fall09.

Syllabus: The schedule in Table 1 is tentative and will be adjusted to the needs of the class; when a topic seems more difficult we shall spend more time on it.

Table 1: Schedule of Lectures 1-14.

	Topics covered	Assignments posted	Assignments due	Reading assignments
1	Class overview, organization System Complexity Sources of Complexity	HW1: exercises 1.1, 1.2, and 1.5		Chapter 1
2	Modularity, Abstraction, Layering Hierarchy, Names Complexity of Computer Systems			Chapter 1
3	Computer Systems versus Other Systems Coping with Complexity The Big Picture			Chapter 2
4	Class Project - Presentation - Discussion		HW1	Chapter 2
5	Fundamental Abstractions Memory, Interpreters, Communication Links			Chapter 2
6	Naming and Layering The Hardware Layer The Software Layer	HW2: exercises 2.1, 2.2, 2.3, and 2.5		Chapter 2
7	Case Study: - The UNIX file system - Naming in UNIX			Chapter 3
8	Modular Sharing Metadata and Name Overloading Addresses and Names		Phase 1 of the project.	Chapter 3
9	Names as a Basic System Component Unique Names Lifetime of Names			Chapter 3
10	User-friendly Names Pitfalls in Naming Case Study: the URL		HW2	Chapter 4
11	Soft/Enforced Modularity Clients and Services Trusted Intermediaries	HW3: exercises 3.2, 3.3, and 3.4		Chapter 4
12	Remote Procedure Call Communication Through an Intermediary			Chapter 4
13	Case Study: - The Internet Domain Name Server (DNS)	HW4: exercises 4.5, 4.6, 4.7 and 4.8		Chapter 4
14	Case Study: - The Network File System (NFS)		HW3	Chapters 4

Table 2: Schedule of Lectures 15-28.

	Topics covered	Assignments posted	Assignments due	Reading assignments
15	Progress Reports for the Project Question and Answer Session for Midterm			Chapters 1-4
16	Midterm			
17	Virtualization; Threads, Virtual Memory;Bounded Buffers. Virtual Links	HW5: exercises 5.3, 5.5, 5.7 and 5.8		Chapters 5
18	Concurrency and Coordination Race Conditions Locks		HW4 + Phase 2 of the project.	Chapters 5
19	Implementation of Locks Enforcing Modularity in Memory Kernel and User Modes			Chapters 5
20	Virtual Memory		Phase 3 of the project.	Chapters 5
21	Threads	HW6: exercises 6.2, 6.7, 6,8 6.10 and 6.11		Chapters 5
22	Threads Synchronization Case Study: Intel x86		HW5	Chapters 6
23	Performance Metrics Reducing Latency Increasing Throughput			Chapters 6
24	Queuing, Basic Models Bottlenecks			Chapters 6
25	Multilevel Memories Spatial and Temporal Locality		Phase 4 of the project.	Chapters 6
26	Scheduling Algorithms, Metrics Case Study: Scheduling the Disk		HW6	Chapters 1-6
27	Discussion of the Project			Chapters 1-6
28	Lessons Learned from the Project The "Big Picture" Revisited			Chapters 1-6