CS 450: Operating Systems, Spring 2017

Instructor	Dr. Suzanne Rivoire (suzanne.rivoire@sonoma.edu)
Lecture	MoWe 10:00–11:50 AM, International Hall 201A
Drop-in office hours	MoWe 4:00–5:00 PM We 12:00–1:45 PM *Office hours are in Darwin 116F. Please knock if the door to 116 is closed.*
Textbook	[required] Thomas Anderson and Michael Dahlin, *Operating Systems: Principles and Practice*, 2nd edition, ISBN 978-0985673529
Prerequisites	Grades of C- or better in CS 315 and CS 252, or consent of instructor.

Catalog Description

Lecture, 4 hours. This course covers the fundamental concepts of operating system design and implementation; the study of problems, goals, and methods of concurrent programming; and the fundamentals of systems programming. Topics include resource-management, process and thread scheduling algorithms, inter-process communication, I/O subsystems and device-drivers, memory management including virtual memory, segmentation, and page-replacement policies. These topics will be covered in theory and in practice through the study of the source-code of a working operating system.

Learning Objectives

For a list of detailed objectives, visit http://rivoire.cs.sonoma.edu/cs450/objectives.html. That list breaks down into 5 major themes:

1. General OS principles: the purpose of operating systems, and the concept of layers of abstraction
2. Concurrency: communication and synchronization between processes
3. Resource allocation and scheduling: ensuring that every process gets its fair share of system resources
4. Virtual memory and virtual machines: the hard work of maintaining useful abstractions for application processes
5. Device management and file systems

Consolidated Syllabus

You may download the course description, objectives, syllabus, and schedule in a consolidated pdf: http://rivoire.cs.sonoma.edu/cs450/syllabus_consolidated.pdf

Exam Dates

Exam 1:	Feb. 22 (Wed.)	In lecture, 10:00–11:50 AM
Exam 2:	Apr. 10 (Mon.)	In lecture, 10:00–11:50 AM
Exam 3 (final):	May 15 (Mon.)	11:00 AM –12:50 PM

Students who have scheduling conflicts on these dates should contact the instructor at the beginning of the semester.

Coursework and Grading

Course Activities

Lecture and Reading

The tentative course schedule shows the topics to be covered. Students are expected to attend all lectures and to get the notes from another student if absent. Students are advised to skim the assigned reading material before each lecture and read more fully after the lecture.

In-class Activities

In-class activities, including quizzes, will be given almost every lecture. Some lectures may include multiple activities. Students' lowest 4 scores on these activities will be dropped from the grade calculation. These activities cannot be made up.

Homework problem sets

Approximately 4 homework sets will be assigned. These assignments may be problem sets, programming projects, or a mix of the two. You may work in groups of up to three students and submit a single solution set for the group.

No late problem sets will be accepted. This policy allows us to discuss their solutions right after the deadline, when applicable.

Exams

Three exams will be given, with the third during the scheduled final exam time. The exams cover the material from lecture, homework sets, activities, and the textbook. Exams will emphasize recent material, although you are responsible for knowing previous material as well. You may bring one 8.5 by 11-inch handwritten sheet of notes to all exams.

Makeup exams will be given only in extraordinary circumstances.

Grading Policies

Grade breakdown

Exams	45%
Homework problem sets	40%
Class activities	15%

Grading scale

93-100%	90-92%	87-89%	83-86%	80-82%	77-79%	73-76%	70-72%	67-69%	63-66%	60-62%	Below 60%
A	A-	B+	B	B-	C+	C	C-	D+	D	D-	F

Up to 3% may be added to your final grade at the instructor's discretion for constructive participation in the class. Constructive participation includes in-class participation; asking good questions via email or during office hours; and doing outstanding or extra work on assignments. No other adjustments of borderline grades will be considered.

Attendance Policy

Your attendance is highly encouraged, and absence from class can affect your grade in the following ways:

- You may miss valuable material in lecture and will need to get notes from another student.
- You may miss graded activities or exams, which can only be made up under extraordinary circumstances.
- A pattern of poor attendance will make it difficult to earn the constructive participation bonus on your final semester grade.

Collaboration Policies

Special note for group work

Your work is the collective responsibility of your group: you will all get the same grade for the assignment, and you will all be held responsible for any violation of the course collaboration policy in the work you submit.

If you start working with a group on a particular assignment but are no longer comfortable sharing this credit or responsibility with one or more of your groupmates, please let me know as soon as possible.

Homework problem set collaboration policy

Problem sets must be the sole work of your group members, and academic misconduct is taken very seriously. You may discuss ideas and approaches with other students and the instructor, but you should work out all details and write up all solutions on your own. **The following actions will be penalized as academic dishonesty:**

- Copying part or all of another group's assignment
- Copying old or published solutions
- Looking at another group's work or discussing another group's work in great detail. You will be penalized if your solution matches another group's solution too closely.
- Showing your group's work or describing your work in great detail to anyone other than your group members or the instructor.

Exam and quiz collaboration policy

Exams and quizzes must be your own work. You are allowed to consult only your own brain, your 8.5x11" handwritten cheat sheet, and other materials specifically permitted by the instructor. Quiz policies will vary and will be announced when the quiz is given. On both exams and quizzes, giving or receiving unpermitted aid will be penalized as academic dishonesty.

Penalties for Academic Dishonesty

Academic dishonesty will be severely penalized; at a *minimum*, you will receive a grade of 0 on the assignment. For more information, see SSU's cheating and plagiarism policy (http://www.sonoma.edu/UAffairs/policies/cheating_plagiarism.htm) and the Dispute Resolution Board website (http://www.sonoma.edu/senate/committees/drb/drb.html).

Course and University Resources

Course Resources

Website

- The course homepage is http://rivoire.cs.sonoma.edu/cs450/.
- The schedule page (http://rivoire.cs.sonoma.edu/cs450/schedule.html) will be regularly updated with links to assignments.
- The resources page (http://rivoire.cs.sonoma.edu/cs450/resources.html) will be updated with links to software tools and helpful resources.

Moodle Gradebook

The course gradebook will be kept on Moodle (http://moodle.sonoma.edu) so that you can check your grades and compute your average at any time. Grades will be posted to Moodle shortly after assignments are returned.

Email List

Course announcements will be sent to your SSU email address, so you should check your email frequently.

University Resources

Disability Accommodations

If you are a student with a disability and you think you may require accommodations, please register with the campus office of Disability Services for Students (DSS), located in Salazar Hall - Room 1049, Phone: (707) 664-2677, TTY/TDD: (707) 664-2958. DSS will provide you with written confirmation of your verified disability and authorize recommended accommodations. This authorization must be presented to the instructor before any accommodations can be made. Visit http://www.sonoma.edu/dss for more information.

University Policies

There are important University policies that you should be aware of, such as the add/drop policy, cheating and plagiarism policy, grade appeal procedures, accommodations for students with disabilities, and the diversity vision statement. Go to this URL to find them: http://www.sonoma.edu/uaffairs/policies/studentinfo.shtml.

CS 450 Learning Objectives

Note: this course's goals and objectives are based on the Operating Systems and Systems Fundamentals areas of the Association for Computing Machinery (ACM)'s Computing Curricula 2013.

General OS Principles

- Explain the objectives and functions of modern operating systems.
- Analyze the tradeoffs inherent in operating system design.
- Describe the functions of a contemporary operating system with respect to convenience, efficiency, and the ability to evolve.
- Discuss networked, client-server, and distributed operating systems and how they differ from single-user operating systems.
- Identify potential threats to operating systems and the security features designed to guard against them.
- Explain the concept of a logical layer.
- Explain the benefits of building abstract layers in hierarchical fashion.
- Describe the value of APIs and middleware.
- Describe how computing resources are used by application software and managed by system software.
- Contrast kernel and user mode in an operating system.

Inter-Process Communication

- Describe how computing systems are constructed of layers upon layers, based on separation of concerns, with well-defined interfaces, hiding details of low layers from the higher layers.
- Describe the mechanisms of how errors are detected, signaled back, and handled through the layers.
- Describe the difference between processes and threads.
- Construct a simple program using methods of layering, error detection and recovery, and reflection of error status across layers.
- Find bugs in a layered program by using tools for program tracing, single stepping, and debugging.

Concurrency

- Describe the need for concurrency within the framework of an operating system.
- Demonstrate the potential run-time problems arising from the concurrent operation of many separate tasks.
- Summarize the range of mechanisms that can be employed at the operating level to realize concurrent systems and describe the benefits of each.
- Explain the different states that a task may pass through and the data structures needed to support the management of many tasks.
- Summarize techniques for achieving synchronization in an operating system (e.g. describe how to implement a semaphore using OS primitives).
- Describe reasons for using interrupts, dispatching, and context switching to support concurrency in an operating system.
- Create state and transition diagrams for simple problem domains.

Resource Allocation and Scheduling

- Compare and contrast the common algorithms used for both preemptive and non-preemptive scheduling of tasks in operating systems, such as priority, performance comparison, and fair-share schemes.
- Describe relationships between scheduling algorithms and application domains.
- Discuss the types of processor scheduling such as short-term, medium-term, long-term, and I/O.
- Compare and contrast static and dynamic approaches to real-time scheduling.
- Discuss the need for preemption and deadline scheduling.
- Identify ways that the logic embodied in scheduling algorithms is applicable to other domains, such as disk I/O, network scheduling, project scheduling, and problems beyond computing.
- Define how finite computer resources (e.g., processor share, memory, storage and network bandwidth) are managed by their careful allocation to existing entities.
- Describe the scheduling algorithms by which resources are allocated to competing entities, and the figures of merit by which these algorithms are evaluated, such as fairness.
- Implement simple scheduling algorithms.
- Use figures of merit to compare alternative scheduler implementations.

Virtual Memory and Virtual Machines

- Explain memory hierarchy and cost-performance tradeoffs.
- Summarize the principles of virtual memory as applied to caching and paging.
- Defend the different ways of allocating memory to tasks, citing the relative merits of each.
- Discuss the concept of thrashing, both in terms of the reasons it occurs and the techniques used to recognize and manage the problem.
- Describe how hardware, VM, OS, and applications are additional layers of interpretation/processing.
- Explain why it is important to isolate and protect the execution of individual programs and environments that share common underlying resources.
- Describe how the concept of indirection can create the illusion of a dedicated machine and its resources, even when physically shared among multiple programs and environments.

Device Management and File Systems

- Identify the relationship between the physical hardware and the virtual devices maintained by the operating system.
- Discuss the advantages and disadvantages of using interrupt processing.
- Explain the use of a device list and driver I/O queue.
- Describe the choices to be made in designing file systems.
- Compare and contrast different approaches to file organization, recognizing the strengths and weaknesses of each.
- Summarize how hardware developments have led to changes in the priorities for the design and management of file systems.
- Summarize the use of journaling and how log-structured file systems enhance fault tolerance.

CS 450: Operating Systems – Spring 2017 Course Schedule

Except for exam dates, all schedule information is tentative. The most recent version of the schedule is online at http://rivoire.cs.sonoma.edu/cs450/schedule.html.

	Monday	Wednesday
Week 1 Jan 23–Jan 27	Introduction Process abstraction Kernel vs. user mode	
	Reading: Ch. 1, 2.1-2.2	
Week 2 Jan 30–Feb 03	More about modes; mode transfer	Mode transfer; syscalls
	Reading: Ch. 2.3-2.5	*Reading: Ch. 2.6*
Week 3 Feb 06–Feb 10	Virtualization Process management *HW 1 assigned*	Process management Fork
	Reading: Ch. 2.10, 3-intro, 3.1	*Reading: Ch. 3.1*
Week 4 Feb 13–Feb 17	Exec and wait I/O redirection	I/O redirection Pipes *HW 1 due*
	Reading: Ch. 3.1-3.2	*Reading: Ch. 3.3-3.4*
Week 5 Feb 20–Feb 24	Concurrency intro Thread programming model *HW 2 assigned*	**EXAM 1**
	Reading: Ch. 4-intro, 4.1-4.3	
Week 6 Feb 27–Mar 03	Synchronization Pthreads lab	Synchronization, continued
	Reading: Ch. 5-intro, 5.1-5.2	Ch. 5.3
Week 7 Mar 06–Mar 10	The bounded buffer problem Condition variables	Read/write locks Bounded buffer lab *HW 2 due*
	Reading: Ch. 5.6.3	*Reading: Ch. 5.4, 5.6*
Week 8 Mar 13–Mar 17	*Spring break - no class*	
Week 9 Mar 20–Mar 24	Deadlock *HW 3 assigned*	Deadlock Simple scheduling schemes
	Reading: Ch. 6.5	*Reading: Ch. 7-intro, 7.1-7.1.3*
Week 10 Mar 27–Mar 31	Scheduling and fairness	Queuing theory basics
	Reading: Ch. 7.1.4-7.1.5	*Reading: Ch. 7.5*
Week 11 Apr 03–Apr 07	Virtual memory intro Segments and pages	Page table organization *HW 3 due*
	Reading: Ch. 8-intro, 8.1, 8.2.2	*Reading: Ch. 8.2-8.3*
Week 12 Apr 10–Apr 14	**EXAM 2**	Page table organization
Week 13 Apr 17–Apr 21	Page table replacement *HW 4 assigned*	Advanced memory management topics
	Reading: Ch. 9.5	
Week 14 Apr 24–Apr 28	Memory management catchup	The file abstraction API Storage devices
		Reading: Ch. 11.1-11.2, 12
Week 15 May 01–May 05	Storage devices Tracking files	Tracking directories *HW 4 due*
	Reading: Ch. 12, 13.2	*Reading: Ch. 13.2, 13.4*
Week 16 May 08–May 12	Bonus topics or catchup	
Finals May 15–May 19	**EXAM 3: Mon. May 15 11:00 AM–12:50 PM**	