CS 450: Operating Systems, Fall 2016

Instructor: Dr. Suzanne Rivoire (suzanne.rivoire@sonoma.edu)
Lecture: MoWe 10:00–11:50 AM, International Hall 201A
Drop-in office hours: We 4:00–5:00 PM
Office hours are in Darwin 116F. Please knock if the door to 116 is closed.
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: Oct. 3 (Mon.) In lecture, 10:00–11:50 AM
Exam 2: Nov. 7 (Mon.) In lecture, 10:00–11:50 AM
Exam 3 (final): Dec. 12 (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 5 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*

<table>
<thead>
<tr>
<th>Category</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exams</td>
<td>45%</td>
</tr>
<tr>
<td>Homework problem sets</td>
<td>40%</td>
</tr>
<tr>
<td>Class activities</td>
<td>15%</td>
</tr>
</tbody>
</table>

**Grading scale**

<table>
<thead>
<tr>
<th>Grade</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>93%</td>
</tr>
<tr>
<td>A-</td>
<td>90%</td>
</tr>
<tr>
<td>B+</td>
<td>87%</td>
</tr>
<tr>
<td>B</td>
<td>83%</td>
</tr>
<tr>
<td>B-</td>
<td>80%</td>
</tr>
<tr>
<td>C+</td>
<td>77%</td>
</tr>
<tr>
<td>C</td>
<td>73%</td>
</tr>
<tr>
<td>C-</td>
<td>70%</td>
</tr>
<tr>
<td>D+</td>
<td>67%</td>
</tr>
<tr>
<td>D</td>
<td>63%</td>
</tr>
<tr>
<td>D-</td>
<td>60%</td>
</tr>
<tr>
<td>Below</td>
<td>59%</td>
</tr>
</tbody>
</table>

Note: Grades are based on a 100-point scale.
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/uaaffairs/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 – Fall 2016 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.

<table>
<thead>
<tr>
<th>Week</th>
<th>Monday</th>
<th>Wednesday</th>
</tr>
</thead>
<tbody>
<tr>
<td>Week 1</td>
<td>Aug 22–Aug 26</td>
<td>Introduction</td>
</tr>
<tr>
<td>Week 2</td>
<td>Aug 29–Sep 02 Process abstraction;</td>
<td>Mode transfer</td>
</tr>
<tr>
<td></td>
<td>Kernel vs. user mode</td>
<td>Ch. 1, 2.1-2.2</td>
</tr>
<tr>
<td>Week 3</td>
<td>Sep 05–Sep 09 Labor Day</td>
<td>Syscalls; Virtualization intro</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ch. 2.6, 2.10</td>
</tr>
<tr>
<td>Week 4</td>
<td>Sep 12–Sep 16 Virtualization; Process</td>
<td>Fork, exec, and wait</td>
</tr>
<tr>
<td></td>
<td>management</td>
<td>Ch. 2.10, 3-intro, 3.1</td>
</tr>
<tr>
<td>Week 5</td>
<td>Sep 19–Sep 23 I/O redirection</td>
<td>Pipes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ch. 3.2</td>
</tr>
<tr>
<td>Week 6</td>
<td>Sep 26–Sep 30 Concurrency intro</td>
<td>Synchronization intro</td>
</tr>
<tr>
<td></td>
<td>Thread programming model</td>
<td>Ch. 4-intro, 4.1-4.3</td>
</tr>
<tr>
<td>Week 7</td>
<td>Oct 03–Oct 07 EXAM 1</td>
<td>Locks</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ch. 5.3</td>
</tr>
<tr>
<td>Week 8</td>
<td>Oct 10–Oct 14 The bounded buffer problem</td>
<td>Read/write locks; condition variables</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ch. 5.6-3</td>
</tr>
<tr>
<td>Week 9</td>
<td>Oct 17–Oct 21 Deadlock</td>
<td>Scheduling: simple schemes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ch. 6.5</td>
</tr>
<tr>
<td>Week 10</td>
<td>Oct 24–Oct 28 Fairness</td>
<td>Multi-level feedback queues</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ch. 7.1.4</td>
</tr>
<tr>
<td>Week 11</td>
<td>Oct 31–Nov 04 Queuing theory basics</td>
<td>Virtual memory intro</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Segments and pages</td>
</tr>
<tr>
<td>Week 12</td>
<td>Nov 07–Nov 11 EXAM 2</td>
<td>Page table organization</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Week 13</td>
<td>Nov 14–Nov 18 Page table organization</td>
<td>Page replacement</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ch. 8.2, 8.3</td>
</tr>
<tr>
<td>Week 14</td>
<td>Nov 21–Nov 25 Memory management catchup</td>
<td>Thanksgiving holiday</td>
</tr>
<tr>
<td>Week 15</td>
<td>Nov 28–Dec 02 The file abstraction API</td>
<td>Storage devices</td>
</tr>
<tr>
<td></td>
<td>Storage devices</td>
<td>Tracking files</td>
</tr>
<tr>
<td></td>
<td>Ch. 11.1-11.2, 12</td>
<td>Ch. 12, 13.2</td>
</tr>
<tr>
<td>Week 16</td>
<td>Dec 05–Dec 09 Tracking files and directories</td>
<td>Filesystem catchup</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ch. 13.2, 13.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>11:00 AM–12:50 PM</td>
</tr>
</tbody>
</table>