San José State University

Department of Computer Science

CS 149, Operating systems, section 3, Fall 2016

Course and Contact Information

Instructor: Jenny Lam
Office location: MacQuarrie Hall 211 (MH 211)
Email: jenny.lam01@sjsu.edu
Office hours: MW 3:00PM-4:00PM or by appointment
Class days/time: MW 7:30AM-8:45AM
Classroom: MacQuarrie Hall 222 (MH 222)
Prerequisites: CS 146 or SE 146 (with a grade of "C-" or better)

Course Description

Fundamentals: Contiguous and non-contiguous memory management; processor scheduling and interrupts; concurrent, mutually exclusive, synchronized and deadlocked processes; files. Substantial programming project required.

Course Goals

- To introduce students to the role of an operating system as a hardware resource manager, and where the OS fits into the software application layer
- To acquaint students with the need to perform memory management, and to explain to them the various memory management techniques and their tradeoffs
- To help students appreciate how the CPU itself is managed by the operating system
- To educate students about the computer deadlock problem, how deadlocks are not unique to the computer system, and attempted solutions to fix the deadlock problem
- To instruct students about processes, their creation, and the software race condition that can happen when multiple processes are run concurrently and perform IPC
- To ensure that students are familiar with the classic IPC problems and how to use semaphores in their software development process to avoid race conditions

Course Learning Outcomes (CLO)

Upon successful completion of this course, students will be able to:

1. Understand the role that the operating system software plays in the management of the various hardware subsystems of the computer system.

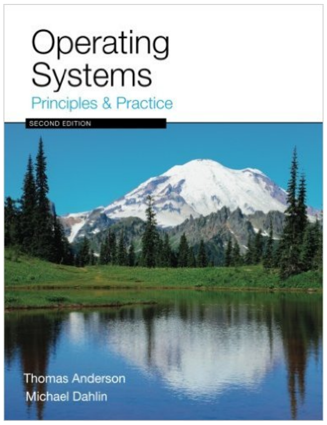
2. Understand locality of memory reference and how it is used to perform effective memory hierarchy management.
3. Understand the various mapping, replacement, and dynamic allocation algorithms for cache and virtual memory management.
4. Understand the alternative CPU scheduling schemes, their tradeoffs, and their applications to other queue processing situations.
5. Appreciate the difficult tradeoffs faced when attempting to deal with the resource deadlock problem and distinguish between the different deadlock prevention and avoidance schemes and understand why and how deadlocks can still happen today.
6. Understand software race conditions, their origin and the problems they can cause, along with knowing how to apply semaphores in software design to solve the race condition problem.
7. Understand the various issues associated with the operating system's role in performing I/O and file management.

BS in CS Program Outcomes (PO)

(a) An ability to apply knowledge of computing and mathematics to solve problems
(b) An ability to analyze a problem, and identify and define the computing requirements appropriate to its solution
(c) An ability to design, implement, and evaluate a computer-based system, process, component, or program to meet desired needs
(i) An ability to use current techniques, skills, and tools necessary for computing practice
(j) An ability to apply mathematical foundations, algorithmic principles, and computer science theory in the modeling and design of computer-based systems in a way that demonstrates comprehension of the tradeoffs involved in design choices
(k) An ability to apply design and development principles in the construction of software systems of varying complexity

Required Texts/Readings

Textbook



Operating Systems: Principles and Practice, 2nd Edition, Thomas Anderson, Michael Dahlin, Recursive Books, ISBN-13: 978-0985673529

Other technology requirements / equipment / material

Students should bring a laptop to every lecture, with sufficient charge to last the entire lecture. The classroom has limited electrical outlets, to which access cannot be guaranteed.

Course Requirements and Assignments

The grade in this course will be weighted as follows:

Participation: although there is no grade incentive, students are expected to attend every class and actively participate.

Assignments: (50%) assignments are assigned roughly every week and will consist of a mix of programming and written problems. Students will be encouraged to work in pairs, but may also choose to work alone. Grading will be done according to the same rubric regardless of the arrangement, and students who worked together will receive the same grade. The two lowest grades will be dropped. Late work is not accepted.

Exams: there will be two in-class exams (15% each) and a comprehensive final exam (20%), to be taken individually. Missed exams cannot be made up except for reasons of illness as certified by a doctor, or documentable extreme emergency. Makeup exams may be oral.

Extra credit problems may be given as part of assignments and exams, the scoring of which may not result in changing the relative weights between assignments and exams.

Letter grade: the final numerical score will translate to the course letter grade according to the following scale:

Percentage	Letter grade
[90-100)	A
[75-90)	B
[60-75)	C
[50-60)	D
[0-50)	F

University Policies

Per University Policy S16-9, university-wide policy information relevant to all courses, such as academic integrity, accommodations, etc. will be available on Office of Graduate and Undergraduate Programs' Syllabus Information web page at http://www.sjsu.edu/gup/syllabusinfo/

CS 149 section 3, Operating systems, Fall 2016, Course Schedule

This schedule is tentative and subject to change during the semester. Changes will be reflected on the course webpage as well as announced during lecture. Students should consult the schedule on the course webpage at http://jennylam.cc/courses/149-f16 regularly for the most up-to-date information.

Week	**Date**	**Topic**
1	W 8/24	Introduction
2	M 8/29	Kernel abstraction
2	W 8/31	Implementing system calls
3	M 9/5	*Labor day*
3	W 9/7	Concurrency and threads
4	M 9/12	Locks
4	W 9/14	Conditional variables
5	M 9/19	Semaphores
5	W 9/21	Deadlocks
6	M 9/26	CPU scheduling
6	W 9/28	Queuing theory
7	M 10/3	flex
7	W 10/5	**Exam 1**
8	M 10/10	Memory protection/translation
8	W 10/12	Memory protection/translation
9	M 10/17	Cache replacement
9	W 10/19	Introduction to I/O
10	M 10/24	File layout
10	W 10/26	File systems
11	M 10/31	Transactions
11	W 11/2	LFS
12	M 11/7	RAID
12	W 11/9	flex
13	M 11/14	**Exam 2**
13	W 11/16	Communication protocols
14	M 11/21	Distributed systems
14	W 11/23	Distributed systems
15	M 11/28	Distributed systems
15	W 11/30	Security
16	M 12/5	Security
16	W 12/7	flex
17	M 12/12	flex
	T 12/20	**Final exam** 7:15AM–9:30AM, MH 222