San José State University
Computer Science Department

CS149, Section 2, Operating Systems, Spring 2020

Course and Contact Information

Instructor:	William "Bill" Andreopoulos
Office Location:	MacQuarrie Hall 416
Email:	william.andreopoulos@sjsu.edu Please use Canvas Messaging and the Discussion Forum
Office Hours:	Monday 16:00-17:00 pm and Wednesday 16:00-17:00 pm
Class Days/Time:	MW 13:30-14:45 pm (Section 2)
Classroom:	Sweeney Hall 100 (Section 2)
Prerequisites:	CS 146 (Data Structures and Algorithms) or SE-146 with a grade of C- or better, or instructor's consent.

Faculty Web Page and MYSJSU Messaging

Course materials such as syllabus, handouts, notes, assignment instructions, etc. can be found on Canvas Learning Management System course login website at http://sjsu.instructure.com. You are responsible for regularly checking with the messaging system through MySJSU at http://my.sjsu.edu (or other communication system as indicated by the instructor) to learn of any updates.

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. Prerequisite: CS 146 or SE 146 (with a grade of "C-" or better). Computer Science, Applied and Computational Math or Software Engineering Majors only; or Instructor Consent.

Course Learning Outcomes (CLO)

Upon successful completion of this course, students will be able to:
   CLO 1 Understand the role that the operating system software plays in the management of the various hardware subsystems of the computer system.
   CLO 2 Understand locality of memory reference and how it is used to perform effective memory hierarchy management.
   CLO 3 Understand the various mapping, replacement, and dynamic allocation algorithms for cache and virtual memory management.

CLO 4 Understand the alternative CPU scheduling schemes, their tradeoffs, and their applications to other queue processing situations.

CLO 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.

CLO 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.

CLO 7 Understand the various issues associated with the operating system's role in performing I/O and file management.

Required Texts/Readings

Textbooks

Remzi H. Arpaci-Dusseau, Andrea C. Arpaci-Dusseau.
Operating Systems: Three Easy Pieces. (*OSTEP*)

This book is available online: http://pages.cs.wisc.edu/~remzi/OSTEP/

Other Readings

•   W. Richard Stevens, Stephen A. Rago. Advanced Programming in the UNIX Environment - 3rd Edition, 2013, Addison-Wesley. (APUE)
•   Robert Love. Linux Kernel Development - 3rd Edition, 2010, Addison-Wesley. (LKD)
•   A. Silberschatz, P. Galvin, and G. Gagne. Operating System Concepts - 9th Edition, 2012, Wiley. http://www.os-book.com/
•   Handouts through Canvas.

Other technology requirements / equipment / material

In this class we will use Virtual Box with Ubuntu as our programming environment for homework assignments – see Canvas for details to install it. We will use a C compiler for programming assignments. Unless otherwise stated, all homework assignments should compile and run using the class VM, which is explained on a Canvas page.

Integrated Development Environment for C - different students use different IDEs. You can choose from vi, nano, visual studio, eclipse, or cLion.

zyBooks – We will also use zyBooks for practicing C programming in-class. You can follow 3 steps to subscribe, as described on Canvas.

Course Requirements and Assignments

SJSU classes are designed such that in order to be successful, it is expected that students will spend a minimum of forty-five hours for each unit of credit (normally three hours per unit per week), including preparing for class, participating in course activities, completing assignments, and so on.

Reading assignments: Readings will regularly be assigned for the next class (see schedule). Slides will be posted under the Canvas modules before the next class.

In-class worksheets: there will be in-class problem solving. These will generally be coding problems (in C) from the reading assignment and/or the homework. The in-class worksheets are a tool for you to learn the material, prepare for exams and practice coding for your future job interviews. These can be done in an IDE of your choice and submitted on Canvas. It is recommended to bring a laptop to class.

The worksheets are graded based on effort and get graded "complete" if a reasonable solution is proposed for each problem, even if it is incorrect (it is understood that the solution might be imperfect). Worksheet submissions are due one week after the class. Please submit what you have by the due date.

Students are encouraged to work in pairs and discuss the worksheet solutions with a partner. Worksheet groups have been created on Canvas, where students can join a group with one other partner to work in pairs. In case two students decide to join a group, the pair will get the same grade for a worksheet they submit together. Students are free to leave a group, or join another group, or work on their own if they prefer. Note working in a pair is optional.

We will take time at the beginning of each class to discuss any difficulties students have in completing the worksheets from previous classes. We will also do code reviews.

While it is fine to discuss the worksheet solutions with your partner, code solutions submitted on Canvas should reflect the students' own efforts in writing the code. *Do not write the code for anyone else. Do not copy code from another source, such as a website, since Canvas automatically checks submissions for plagiarism from multiple online sources.*

In-class iClicker questions: The iClicker questions are in the form of multiple choice and true-false questions. All students are expected to participate with iClicker. Credit is given for participation and it is not necessary to get the correct answer to get credit. Please install iClicker on your phone (app) or laptop (iclicker.com) following these instructions: http://www.sjsu.edu/ecampus/teaching-tools/iclicker/

Homework assignments: Programming assignments will be assigned. Students are encouraged to work in pairs and discuss the assignment solutions with a partner. Assignment groups have been created on Canvas, where students can join a group with one other partner to work in pairs. In case two students decide to join a group, the pair will get the same grade for a worksheet they submit together. Students are free to leave a group, or join another group, or work on their own if they prefer. Note working in a pair is optional.

More information will be given at the time of the first programming assignment. Never copy any code you find on the web. Penalty for late submission 5% for every 3 days up to 15 days, after that no submission will be accepted. Never email your assignments, always upload to Canvas. Oral examination might be requested.

Midterm exams: There will be two Midterm exams during the semester.

Final Examination or Evaluation

Final exam: One final cumulative exam.

The exams will contain multiple choice questions, true/false and short answer questions. Exams are *closed book*, final exam is comprehensive. No make-up exams except in case of verifiable emergency circumstances.

Discussion Forum on Canvas

This term we will be using the Discussion Forum on Canvas for class discussion. The system is highly catered to getting you help fast and efficiently from classmates, the TA, and myself. Rather than emailing questions to the teaching staff, I encourage you to post your questions on the Discussion Forum on Canvas.

Extra credit opportunity

A student can volunteer to present in-class her solution for an assignment or a worksheet. Students have to express interest in presenting (by messaging or speaking with the instructor). An assignment or hands-on can only be reviewed once. A review lasts for 20 minutes max. These will take the form of code reviews, where the student walks us through her code solution and we discuss the proposed solution and if there are better ways to solve the problem. Extra credit of 1% for a student who reviews her solution for an assignment or a worksheet in class.

Determination of Grades

Final Grade is based on:

40% Assignments
20% Midterms (10% each)
30% Final
9% In-Class worksheets
1% iClicker questions

Grade	Points	Percentage
A plus	960 to 1000	96 to 100%
A	930 to 959	93 to 95%
A minus	900 to 929	90 to 92%
B plus	860 to 899	86 to 89 %
B	830 to 859	83 to 85%
B minus	800 to 829	80 to 82%
C plus	760 to 799	76 to 79%
C	730 to 759	73 to 75%
C minus	700 to 729	70 to 72%
D plus	660 to 699	66 to 69%
D	630 to 659	63 to 65%
D minus	600 to 629	60 to 62%

Communication with the instructor

Students should post any general questions on the Canvas Discussion Forum, where the entire class can benefit from the responses. To email the instructor, please use Canvas messaging rather than a direct email address, since this helps the instructor to organize and keep track of all course-related electronic communication. The instructor responds to course-related electronic messages in this priority: Discussion Forum, then messages sent through Canvas.

Graders/TAs

Aleksandra Khovina
aleksandra.khovina@sjsu.edu

Shivangi Jain
shivangi.jain@sjsu.edu

Classroom Protocol

Attendance is highly recommended. Please avoid disturbing the class: turn-off cell phones (or put them on silent or vibrate mode), no text messaging in class or during exams, **no taking pictures and video**, avoid coming late. You are not allowed to publically share or upload material for this course such as exam questions, lecture notes, or solutions without the instructor's consent.

Regrading Procedure

In the event that a student requests a regrade of a question on a homework or exam, please follow the procedure described next. A print out (not email) indicating the specific question(s) should be provided to the instructor, along with a note describing the issue and the reason for the regrading request. Please also indicate on the note your name, course section, assignment and question number.

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](http://www.sjsu.edu/gup/syllabusinfo/) at http://www.sjsu.edu/gup/syllabusinfo/

CS149: Operating Systems, Spring 2020

The schedule is subject to change with fair notice.

Course Schedule

Lectures	Date	Topic
1	1/27	Introduction
2	1/29	Review C and the command line
3	2/03	Review C and the command line
4	2/05	Processes
5	2/10	Process API
6	2/12	Direct Execution
7	2/17	System calls with File I/O
8	2/19	Interprocess Communication, Sockets, Pipes
9	2/24	Signals
10	2/26	CPU Scheduling
11	3/02	Multilevel CPU Scheduling
12	3/04	**Midterm 1**
	3/09	Hard Disks
13	3/11	Files and Directories

Lectures	Date	Topic
14	3/16	File System Implementations
15	3/18	Address Space
16	3/23	Memory API
17	3/25	Free-Space Management
	3/30	*Spring recess*
	4/01	*Spring recess*
18	4/06	Paging
19	4/08	TLB
20	4/13	Swapping Policies
21	4/15	Thread API
	4/20	**Midterm 2**
22	4/22	Locks
23	4/27	Lock-based concurrent DS
24	4/29	Multi-Core CPU programming
25	5/04	Condition variables and semaphores
26	5/06	Concurrency bugs
27	5/11	Review for exam
		Final exam for Section 2: Monday, May 18, 12:15-14:30