San José State University
Computer Science Department

CS149, Operating Systems, Summer 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
Class Days/Time:	MW 9:00-11:00 am
Classroom:	Online
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 Canvas messaging system to learn of any updates. You should modify the Canvas settings for notifications of announcements and discussion forum postings to be sent to you.

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. It is understood that a worksheet solution might be imperfect or have a few errors.

Worksheet submissions are due one week after the class. The worksheet submission page closes after one week. Please submit what you have by the due date.

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.

Homework assignments: Programming assignments will be assigned.

More information will be given at the time of the first programming assignment. Penalty for late submission 5% for every 3 days up to 15 days; after 15 days no submission will be accepted and the submission page will be closed. Never email your assignments, always upload to Canvas.

Students are encouraged to work in pairs and discuss the worksheet/assignment solutions with a partner. Students can join a group with one other partner to work in pairs. If two students join a group, the pair will get the same grade for a worksheet/assignment they submit together. Only one group member should submit a worksheet/assignment (to avoid double grading) and include both members' names on the submission. If both group members submit a worksheet/assignment, causing double grading, there will be a penalty. 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.

While it is fine to discuss the worksheet/assignment solutions with your partner within your group, code solutions submitted on Canvas should reflect the students' own efforts in writing the code. *Do not write the code for anyone else. Never copy any code you find on the web. Do not copy code from another source, such as a website, since Canvas automatically checks submissions for plagiarism from multiple online sources.* Oral examination might be requested.

iClicker participation during class: 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/

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

Final exam: One final cumulative exam.

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

Discussion Forum on Canvas

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 (via Zoom) 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:

50% Assignments
20% Midterms (10% each)
20% Final
9% In-Class worksheets
1% iClicker participation

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

The course instructor receives a very large volume of emails during the current remote teaching situation and he can not respond to every message received. Therefore, it is essential that you follow the correct channels for communication. Questions should preferably be done during the regular class meeting time via Zoom. The instructor does not write or reply to emails after normal business hours, on weekends or holidays.

The instructor responds to course-related electronic messages by priority of Discussion Forum postings, then Canvas messages, and then direct emails:
1) Students should post questions on the Canvas Discussion Forum, where the entire class can read and benefit from the responses. The discussion forum postings may also be discussed in class.
2) Students should preferably use Canvas messaging rather than direct email, since this helps the instructor to organize and keep track of all course-related electronic communication. The instructor will often re-post Canvas messages and responses to the discussion forum (see #1).
3) Private emails sent to the instructor's gmail will be responded to after the discussion forum (see #1) and canvas messages (see #2) have been cleared. The instructor will often re-post these emails and responses to the discussion forum.

Announcements that concern everyone, such as reminders about due dates or class policy, will be posted under Announcements on Canvas.

Reviewing code for the assignments and technical trouble-shooting should preferably be done during the regular class meeting time via Zoom. Instead of sending the instructor your entire code via email, come to class and discuss it there. If the course instructor helps you to fix code bugs, this does not guarantee you will receive an excellent grade for an assignment. The assignment grade is based on a scoring rubric with additional criteria, such as whether the requirements are satisfied, tests pass, and overall code quality.

Graders/TAs

Aleksandra Khovina
aleksandra.khovina@sjsu.edu

Siddartha Thentu
siddartha.thentu@sjsu.edu

Classroom Protocol

Attendance (via Zoom) is highly recommended. 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

Grades assigned are final, unless there was an error in the grading. In the event that a student wants to request a regrade of a homework or test, please follow the procedure described next. You should fill out the "Regrade request" form on Canvas. A request for a regrade is not a technique to drum up a few more points. If the course instructor thinks a component was scored too highly the first time, it may be lowered in a regrade. *The overall grade may increase, decrease, or stay the same after a regrade request*.

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/

CS149: Operating Systems, Summer 2020

The schedule is subject to change with fair notice.

Course Schedule

Classes	Topic
1	Introduction
2	Review C and the command line
3	Processes / Process API
4	Direct Execution
5	System calls with File I/O
6	Interprocess Communication, Sockets, Pipes
7	Signals
8	CPU Scheduling / Multilevel CPU Scheduling
9	**Midterm 1** / Hard Disks
10	Files and Directories
11	File System Implementations
12	Address Space / Memory API
13	Free-Space Management
14	Paging
15	Swapping Policies
16	**Midterm 2** / Thread API
17	Locks / Lock-based concurrent Data Structures
18	Condition variables and semaphores
19	Concurrency bugs / Advanced Locks
20	**Final exam**