

Course Syllabus

Instructor:

Justin Bradley 290 Schorr
Office hours: By appointment via Zoom
Email: justin.bradley@unl.edu
Phone: (402) 472-5072

TAs:

If you request help from the TAs please briefly describe your problem and include files you'd like them to look at.

Karthik Gorla
Office hours: Wednesdays and Thursdays 3:30-4:30
Email: kgorla@cse.unl.edu

Tyler Barker
Office hours: Wednesdays and Thursdays 10:00-11:00
Email: tyler.barker@huskers.unl.edu

1 Course Overview

This course will introduce students to key concepts of modern operating systems. These concepts include system organization, processes, threads, process management, process and thread synchronization, memory management, and storage management.

Topics covered: Organization and structure of operating systems. Processes and concurrent programming. Inter-process communication. Process synchronization using reads/writes, semaphores, and monitors. CPU scheduling. Deadlocks including detection, avoidance, prevention and recovery. Intro to real-time systems. Memory organization and management including paging, segmentation, virtual memory, and page replacement algorithms. File system concepts and structure. Protection and security. Topics are conveyed through lectures and a series of hands on programming assignments.

2 Course Requirements

Prerequisites: A grade of "P" or "C" or better in CSCE 230, CSCE 230H or CSCE 231 and CSCE 310, CSCE 310H, CSCE 311, SOFT 260, SOFT 260H or RAIK 283H.

Text: Bic, Lubomir. [Operating Systems Principles. \(Links to an external site.\)](#) zyBooks, 2020, zyBook ISBN: 978-1-394-06717-6

To access and subscribe to the book follow these instructions:

1. Click any zyBooks assignment link in your learning management system (e.g., Canvas) (Do not go to the zyBooks website and create a new account)

2. Subscribe

3 Class Details

This course is “online” meaning there is no in-person component, and no required regular meeting time. Asynchronous video lectures and readings in the book will be used to introduce concepts to students. Programming assignments and associated tips will be used to ensure you can get started on them and hopefully be successful.

Final exam: Anytime 00:00-23:59 Wednesday, May 11, 2022

3.1 Differences between 451 and 851

CSCE 851 students will take a different, more comprehensive final exam. Otherwise, assignments between 451 and 851 are identical.

4 Website, Communication, and Other Logistics

4.1 Canvas

I will use Canvas to disseminate materials, host the class schedule, maintain all links, etc. To (hopefully) simplify everyone’s life the front page of the course is a page called “Main Schedule”. This page will hold a running list of announcements, links to all resources, assignments, as well as a schedule.

You need to check this page regularly. Canvas, depending on your settings, may send you a notification that the page has been updated (but it will look like raw HTML). You’ll then need to go check it. I do it this way to reduce the number of Canvas announcement emails you receive.

Special Note: the “Main Schedule” really is the main schedule – meaning it supersedes all other dates in Canvas, CSE Handin, or elsewhere.

4.2 Other Resources

Other external materials, links, etc. will be used as follows (all links to these will be in the “Main Schedule” page in Canvas):

- **Piazza** will be used for all online discussion of homework, programming assignments, and logistics.
- **YouTube** will be used to host lecture videos.
- **Microsoft OneDrive** will be used to host programming assignments and other handouts.
- **Zoom** will be used for live office hours and/or personal meetings.

- **zyBooks** will be used as our book. Readings and homework assignments will be given from this book, and homework assignments, based on the readings, will be given. You will have to subscribe to the book (it is much cheaper than the previous book).
- **CSE Handin** will be used to turn in programming assignments.

5 Policies

5.1 Grading Policies

This course will consist of 6 (graded) programming assignments, weekly reading check assignments, and a final exam. The grade breakdown will be:

- Programming Assignments: 60%
 - PA0 – 5%
 - PA1 – 11%
 - PA2 – 11%
 - PA3 – 11%
 - PA4 – 11%
 - PA5 – 11%
- Exams: 20%
 - Final Exam: 20%
- Homework - 20%
 - 7 reading/homework assignments – $20\%/7 = \sim 2.86\%$
- Extra Credit:
 - Evaluation: 2% bump to final percentage (only if >80% of class fills out evaluation)

5.1.1 Late Work Policy

Late work will not be accepted. I *will* make exceptions for [UNL-approved reasons](#)[Links to an external site.](#) I *may* make exceptions for other reasons if you discuss it with me before the deadline. If you have an issue and need an extension, please talk with me.

Explanation: I have tried several variations on accepting late work. Accepting late work means (much) more work for me and the TAs, and this class already has a high grading burden. My anecdotal evidence suggests the rate of students turning things in is about the same. Not accepting late work is simplest and encourages students to get started on assignments earlier.

5.1.2 A very important note on grading of programming assignments:

- Programming assignments will be graded on a 100 point scale.

- Your program *must* compile and execute on the CSE servers (cse.unl.edu) - else you get 0/100. Make sure you check this prior to handing in.
- All programming assignments must be handed into [CSE web handin](#)~~Links to an external site.~~ - else you get 0/100.
- Each programming assignment will indicate precisely how it will be scored. **Most** of the time points are gained by passing the test suites. Test suites are often broken into different parts so that you can get at least partial credit for partial implementations. You won't get any points for tests that you don't pass. This means there's no such thing as "being close" or "almost works."
- Despite the previous point, I also reserve the right to take off points for poor programming practices: poor commenting, function names, variable names, magic numbers, etc.

5.1.3 Explanations

Students have a variety of reactions to these policies. Here are some common ones, and associated explanations:

- "*That's stupid!*": When you develop code in industry you don't get "credit" if you break the nightly build (i.e., your code doesn't compile), or your program simply doesn't do what it is supposed to do (i.e., test suite fails). And most businesses have firm deadlines to meet for product releases.
- "*That doesn't accurately reflect what I learned!*": I'm in the business of preparing our students to be strong employees. As 4th year undergraduates, or graduate students, good programming practice is simply an expectation of a computer scientist, software engineer, or computer engineer. Writing programs that meet specifications is the *bare minimum* any computer scientist, software engineer, or computer engineer can do. In most PAs you will have access to the test suites or an approximate set of test suites, so it should be straightforward to test all your code, and make sure it compiles and runs on the CSE servers.
- "*Why not partial credit?*" or "*I was so close...*" You will get the indicated points for passing various tests within the test suite. If your code doesn't pass a test you get 0 points for that particular test. There are several reasons for this:
 - I don't know how to give partial credit. If the output of your code doesn't match the required output, I (or the TA) would need to look at your code and make an assessment of how "close" you were to being correct. I've no idea how to do that fairly or efficiently.
 - To my knowledge, there is no agreed upon metric for determining how "close" code is to being "correct."

Bottom line: write *good* code that fulfills the specifications (i.e., passes all the tests) and you'll get all the points.

5.1.4 Grading Appeals Policy

Sometimes you do your very best and something doesn't go right. You forgot a "\n" character in a printed statement, etc. In these minor cases, where the logic is correct, but the output may not be quite right, please write me an "appeals" email describing the scenario, submitting your new code, and I will, at my discretion, re-grade the assignment with a 10% penalty. All appeals must be filed within a week of grades being posted.

5.1.5 Grade Assignment

Grades will then be assigned using the following table:

Grade Percent

A	[93-100)
A-	[90-93)
B+	[87-90)
B	[83-87)
B-	[80-83)
C+	[77-80)
C	[73-77)
C-	[70-73)
D+	[67-70)
D	[63-67)
D-	[60-63)
F	[0-60)

5.2 Class Policies

1. Put your name, course, and section on everything you turn in. Sometimes we print things out for grading. Also, this is generally just good practice.
2. Exchange of ideas and techniques is **highly** encouraged but **your work must be your own**. If someone helps you, please give them credit in your code (even if it's the TA). Myself and the TAs will follow up with suspicions of academic dishonesty in accordance with department and university policy (see below).
3. Students have one week from time of grade posting to challenge a grade.
4. If you have *technical content* questions please follow this order of operations for getting help:
 - a. Look on Piazza and check for similar questions.
 - b. Post your question on Piazza – then everyone benefits from the answer!
 - c. Check the internet for similar questions or concepts.
 - d. Attend the TAs' office hours.
 - e. Email the TA asking your question or requesting a Zoom session.

- e. Email me asking your question or requesting a Zoom session.
- 5. If you have *personal questions* about logistics or similar please skip straight to contacting me by email explaining the situation and your question.
- 6. Please use professional email communication. A salutation, description, request, closing, and concise subject line are appropriate for professional communication.
- 7. I'm particularly concerned with the mental and emotional distress our current circumstances could have on each of us. Don't hesitate to reach out and let me know how things are going. Large doses of patience are in order during the global pandemic.

5.3 CSE Policies

- 1. The CSE Department has an [anonymous contact form](#)Links to an external site. that you may use to voice your concerns about any problems in the course or department if you do not wish to be identified.
- 2. CSE Department policy dictates that students in CSE courses are expected to regularly check their email so they do not miss important announcements.
- 3. Consider the [Student Resource Center](#)Links to an external site. in Avery 12 if myself or the TAs are not available, or you otherwise need help.
- 4. All homework assignments, quizzes, exams, etc. must be your own work. No direct collaboration with fellow students, past or current, is allowed unless otherwise stated. The Computer Science & Engineering department has an [Academic Integrity Policy](#)Links to an external site.. All students enrolled in any computer science course are bound by this policy. You are expected to read, understand, and follow this policy. Violations will be dealt with on a case by case basis and may result in a failing assignment or a failing grade for the course itself. The UNL College of Engineering also has an [academic integrity policy](#)Links to an external site. you should read and understand.

5.4 UNL Policies

The University has put together a regularly updated [page of course policies and resources](#)Links to an external site.. Please make sure you are familiar with these policies.