Introduction to Computer Systems

Syllabus

Class Resources

- Canvas: https://canvas.northwestern.edu/courses/137178
  Everything from the course goes here: course content, grades, Zoom URLs, and class recordings.

- Campuswire: https://campuswire.com/c/G76A6D558/feed
  Course discussion and questions. **All questions should go to Campuswire rather than to email.** We will enroll you.

- Office hours: See Canvas

Office hours will be set during the first week based on student input. Our goal is to have every student be able to attend at least one office hour per week.

Instructor

Branden Ghena
branden@northwestern.edu

Teaching Assistance

Drake Han (TA)
HaiyangHan2022@u.northwestern.edu

Huaxuan Chen (PM)
HuaxuanChen2023@u.northwestern.edu

Neil Vakharia (PM)
neilvakharia2021@u.northwestern.edu

Jacob Tucker (PM)
jacobtucker2023@u.northwestern.edu

Atishay Saraogi (PM)
atishaysaraogi2023@u.northwestern.edu

Seth May (PM)
sethMay2022@u.northwestern.edu

Spencer Colton (PM)
spencercolton2021@u.northwestern.edu

Anthony Rotyman (PM)
anthonyroytman2021@u.northwestern.edu

Peter Ha (PM)
peterha2023@u.northwestern.edu

Location and Time

Lectures:	Tuesdays and Thursdays, 2:00-3:20pm Central, Online "Synchronous Remote" Model Recordings available afterward
Midterm Exam 1:	Thursday, April 29th, during lecture time
Midterm Exam 2:	Thursday, June 8th, during lecture time

Prerequisites

Required CS 211 or equivalent

- Experience with C or C++
- Some experience with programming in a Unix environment

CS 213 is a **required core course** in the Computer Science curriculum in both McCormick and Weinberg. It is also a required course for CS minors in both schools. 213 can also be taken for credit within the Computer Engineering curriculum. 300-level computer systems courses generally have 213 as a prerequisite.

Textbook

Required Textbook:
Computer Systems: A Programmer's Perspective, ***Third Edition****,*
Randal E. Bryant and David R. O'Hallaron,
Prentice Hall, 2015, (ISBN-13: 978-0134092669, ISBN-10**:** 013409266X)

- Details on http://csapp.cs.cmu.edu/3e/students.html
- **Make sure you have the third edition of the book**. This edition is the first to focus on the 64 bit operation of the machine, which we will make extensive use of in this course.
- If you buy a non-U.S. version, acquire a pdf through some means, etc, please be aware that these can have differences from the U.S. version.
- This class is eligible for the Books for Cats program; the textbook is available via library reserves for eligible students. Please contact Brian Drabik (b-drabik@northwestern.edu) for eligibility questions.

Objectives, framework, philosophy, and caveats

This course introduces the lower levels of computer systems. We will peel back the abstractions you have grown accustomed to from CS111 and CS211, and look “under the hood”, so to speak, to see how they work. In doing so, we will explore the hierarchy of abstractions and implementations that comprise contemporary computer systems.

The goals for doing so are two-fold. First, this will provide you with an overview of “what’s out there” in the fascinating world of computer systems, which you can flesh out in our upper-level computer systems courses. Second, and more important in the long term, your familiarity with the lower levels of systems will allow you to investigate and resolve issues that arise when the upper levels break down due to bugs or performance issues.

This is a learn-by-doing kind of class. You will write pieces of code, compile them, debug them, disassemble them, measure their performance, optimize them, etc.

The specific computer architecture we will focus on in this class is the 64 bit Intel/AMD x86 architecture, which is used in virtually all supercomputers, clouds, clusters, servers, desktops, and laptop/notebook computers today.[1] The specific operating system we will use is Linux, which is used in most supercomputer, cloud, cluster, and server environments, and is the operating system of Android smartphones and ChromeBooks. The specific programming toolchain we will use is GCC (and GDB), which is an extremely widely used core toolchain on pretty much all platforms, except Windows. The ideas and concepts embodied in this architecture, operating system, and programming toolchain are commonly found in others.

Lectures

Lectures will be streamed live on Zoom at the regularly schedule course time. Links to the Zoom meetings are available on the “Zoom” tab in Canvas.

Attendance is strongly encouraged (I love participation!), but not mandatory. If you cannot attend live lectures for whatever reason (time zone, connectivity, etc.), lectures will be recorded and automatically made available on Canvas under the “Panopto” tab.

I strongly encourage keeping up with the course lectures. Much of the difficulty in CS213 comes from the new concepts, which we take care to introduce in lecture.

[1] The 64 bit x86 architecture is also called “x86_64” and just “x64”. We may also touch on the ARM architecture used heavily in smartphones and the relatively new RISC-V architecture. If this doesn’t make sense to you yet, don’t worry about it, this course will teach you.

Portions of this class will be recorded by the instructor for educational purposes. These recordings will be shared only with students enrolled in the course and will be deleted at the end of the quarter.

Other Ways of Getting Help

Your instructors, TA, and PMs will also have regularly scheduled office hours and be available by appointment if these do not work. We will schedule office hours in the first week to maximize opportunities to attend.

We will use an online discussion group on Campuswire as well. We will enroll you. The link is on the course web page. The intent is to have multiple venues for discussion with different styles so that all students feel comfortable participating. If you have a question, answer, or comment, please put it forward. We will try our best to answer.

Class Servers

Machines at Northwestern will be available for remote login. You will have access to several server machines that can support many users simultaneously, and we expect most students will use those servers. We will test your labs on those machines. You should also be able to work on labs on your own machine provided it is running a reasonably recent Linux.[2]

Labs

This course will have four programming labs. Their goal is to make you apply the concepts you have learned and understand them more deeply. Labs should be done in groups of two. **Start looking for a partner on day one.**

Homeworks

We will give you several graded homework assignments, give you some time to work on them and hand-in the results, and then provide solutions. Homeworks are to be done **individually** and are important for preparing for exams.

[2] If you would like to do this, but your machine uses Windows or MacOS, you can install virtualization software, and then install Linux in a virtual machine. We typically use VMware for this (Player on a Windows box, Fusion Pro on a Mac), but there are other tools. Ubuntu is a reasonably good choice of Linux for this purpose.

Exams

We will have two *synchronous* exams.

Exams will be electronic (via Canvas) and open notes, but general Internet access will be forbidden. No communication with anyone other than course staff will be allowed during the exam. Be aware: we have a number of tools at our disposal to detect improper collaboration, and we will pursue any we find to the fullest extent possible. To avoid any temptation, we ask that you take exams alone, with no one else in the room.

To account for multiple time zones, we may make a second exam time available for students who are affected based on their input. If you are unable to take an exam during the normal class time, please let the course staff know as soon as possible.

Grading

50 %	Programming labs (4 labs, 12.5% each)
20 %	Homeworks (4 homeworks, 5% each)
15 %	Midterm exam 1 (covers first half of the course)
15 %	Midterm exam 2 (covers second half of the course)

For some of the programming labs, extra credit may be possible.

Your score in the course is the weighted average of your scores on each of the components. You can view all currently graded material, and your score, at any time on Canvas. Final grades are based on the course score (the weighted average), with the basic model being that the 90s are A territory, 80s are B territory, and so on. This model will be adapted toward lower thresholds if necessary based on overall class performance. That is, this is NOT a curved class.

The instructors ultimately assign scores and grades in consultation with the TA and PMs. If you have a problem with a score on an assignment/exam or your grade, you are welcome to bring it up with them or the instructors, but only the instructors are empowered to change grades.

Late Policy

For each calendar day after the due date for a lab, 10% is lost. After 1 day, the maximum score is 90%, after 2 days, 80%, etc, for a maximum of 4 days.

Academic Integrity

Collaboration is a really good thing, and we encourage it. On the other hand, cheating is a very serious offense, which carries serious consequences. It is OK to meet with colleagues, form study groups, discuss assignments with them, compare alternative approaches, go over examples from textbooks or other sources. **But it is never ok to share code or homework solutions, or even to see each other's code or solutions.**

What you turn in must be your own work. Copying (or even studying) code, solution sets, etc., from anywhere (e.g., other people, web, GitHub) is strictly prohibited. Be aware that we use a number of tools to detect and discover integrity violations. If you discuss your work with other people, please acknowledge them by listing their names in your submission. It is also forbidden to share, post, or otherwise publicize course materials. This includes (but is not limited to) homeworks, exams, solutions, or your own submissions. This extends even after the quarter ends; course material remains private information which you may not share or reproduce.

It is the responsibility of every student in this class to be familiar with and to adhere to the Academic Integrity Policies of Northwestern University and the McCormick School of Engineering. Any suspicion of violation of these policies will be reported immediately to the Associate Dean for Undergraduate Studies. If you are in doubt whether your actions constitute a violation of the above policies, ask the instructor (preferably before doing what you are unsure about).

Accessibility / ANU

Any student requesting accommodations related to a disability or other condition is required to register with ANU (accessiblenu@northwestern.edu; 847-467-5530) and provide professors with an accommodation notification from AccessibleNU, preferably within the first two weeks of class. All information will remain confidential.

Should you need them, additional campus resources are available, including, but not limited to:

- Accessible NU www.northwestern.edu/accessiblenu/
- CAPS www.northwestern.edu/counseling/index.html
- Student Enrichment Services www.northwestern.edu/enrichment/

I believe in providing reasonable accommodations that allow for full access to learning for all. Please contact me if there is anything that I should be aware of that might have an impact on your participation in this course (documented disability, language challenges, absences for religious observations, etc.).

Diversity and Inclusion

I consider this classroom to be a place where you will be treated with respect, and I welcome individuals of all ages, backgrounds, beliefs, ethnicities, genders, gender identities, gender expressions, national origins, religious affiliations, sexual orientations, ability—and other visible and nonvisible differences. All members of this class are expected to contribute to a respectful, welcoming, and inclusive environment for every other member of the class.

Tentative Schedule

(Subject to change, especially the lecture schedule, less so the homeworks/labs)

Lecture	**Date**	**Topics**	**Readings**	**Homework/Labs**
1	4/01 Th	Overview	Ch. 1	Homework 1 out
4/05 is the last day for adding courses or changing sections.				
2	4/06 T	Integer Representations	Ch. 2.1, 2.2	Data Lab out
3	4/08 Th	Integer Operations	Ch. 2.3	
4/09 is the last day for tuition adjustment based on enrollment changes.				
4	4/13 T	Floating Point	Ch. 2.4, 2.5	Homework 1 in / Homework 2 out
5	4/15 Th	Machine Basics	Ch. 3.1-3.4	
6	4/20 T	Arithmetic Operations and Conditions	Ch. 3.5, 3.6.1, 3.6.2	Data Lab in / Bomb Lab out
7	4/22 Th	Control Flow	Ch. 3.6.3-3.6.8	
8	4/27 T	Procedures	Ch. 3.7	Homework 2 in
9	4/29 Th	**Midterm Exam 1**		
10	5/04 T	Pointers and Arrays	Ch. 3.8	Homework 3 out
11	5/06 Th	Structured Data	Ch. 3.9, 3.12	
5/07 is the last day to drop a course for the Spring term.				
12	5/11 T	Buffer Overflows	Ch. 3.10.1-3.10.4	Bomb Lab in / Attack Lab out
13	5/13 Th	Memory Hierarchy	Ch. 6.1-6.3	
14	5/18 T	Caches	Ch. 6.4	Homework 3 in / Homework 4 out
15	5/20 Th	Cache Performance	Ch. 6.5-6.7	
16	5/25 T	Concurrency	Ch. 12.3	Attack Lab in / SETI Lab out
17	5/27 Th	Virtual Memory	Ch. 9.1-9.7	
18	6/01 T	Processes	Ch. 8.2, 8.4, 12.1	Homework 4 in
19	6/03 Th	**Midterm Exam 2**		
	6/08 T	-- (exam week) --		SETI Lab in