

Syllabus

Class Information

- Class Name: CSCI 356 Introduction to Computer Systems
- Units: 4
- Session: Spring 2023
- Instructors: Marco Paolieri
- Teaching Assistants: see the [list of TAs](#)
- Office Hours: see the [calendar](#)

Course Description

This course covers computer systems hardware organization and the programmer interface with the goal of improving students' abilities to reason about the execution of their programs, write system software, and enhance the performance of the programs they write. This course will also serve as a basis for other systems courses, such as CSCI 350 (Operating Systems), CSCI 450 (Computer Networks), or CSCI 457 (Computer Systems Organization), EE 451 (Parallel and Distributed Computation).

Concepts include information representations, assembly language programming and the nuances of the x86 instruction set, the memory hierarchy including DRAM and cache, address translation and virtual memory, instruction execution on a modern processor with out-of-order execution, along with compilation and system software concepts such as linking and heap management.

Five programming assignments will provide the opportunity for students to apply the skills they have gained in the context of real applications. Many assignments will involve reverse engineering machine code or attempting to optimize performance of certain code kernels. Along the way use of debuggers and other binary tools will be taught and practiced.

The fundamental concepts and skills learned in this class underpin the modern state-of-the-art of computer system organization and will be the basis for future courses in operating systems, compilers, networks, and computer architecture.

Learning Objectives

Below are the specific, measurable skills a student will demonstrate by the end of the course. These objectives will be both taught and assessed in the course and are aligned with the assignments, assessments and learning materials.

1. Translate bit strings to numbers using unsigned, 2's complement, and IEEE standard floating point representation systems.
2. Reverse engineer machine code and assembly code to a behavioral (high-level) descriptions.
3. Translate high-level software code to corresponding x86 instruction sequences.
4. Assess the cache performance of a system given its memory/cache specifications and a specific address trace.
5. Identify various software vulnerabilities and how they may be exploited.
6. Manually simulate address translation as a means of understanding hardware and software components that do likewise.
7. Experiment to determine efficient storage (specifically heap memory) allocation strategies.
8. Organize code and use compiler settings to achieve enhanced performance on specific processor architectures.

Prerequisites

- CS 104L (Data Structures and Object Oriented Design)

Co-Requisites

None

Concurrent Enrollment

Courses that must be taken simultaneously.

Recommended Preparation

None

Course Materials

- All content will be provided on our website. We will post PDF slides before each lecture.
- The required textbook for the course is *Computer Systems: A Programmer's Perspective*, by Bryant and O'Hallaron (3rd edition, Pearson, 2015, ISBN 978-0134092669).

Course Websites

1. **Primary website:** All course assignments, content, office hours information, will be posted at usc-cs356.github.io.
2. **Q&A website:** For Q&A and announcements, we will use <https://piazza.com/usc/spring23/csci356/home>. All announcements regarding assignments, lectures, exams, etc. will be made via Piazza. It is your responsibility to check this site often.
3. **Grades:** Blackboard (blackboard.usc.edu) will be used to release assignment and exam grades; Gradescope (www.gradescope.com) will be used to handle regrade requests.

Technological Proficiency and Hardware/Software Required

This course requires you have access to a computer/laptop able to run VS Code and to connect to our programming server through SSH. Please visit the course website for more information.

You will also be provided access to private GitHub repositories where you must commit and push your code submissions. You are expected to be familiar with basic git and GitHub skills.

Required Readings and Supplementary Materials

The following textbook is **required**. We recommend you read the sections listed on the course schedule for the corresponding week BEFORE attending the first lecture of that week.

- *Computer Systems: A Programmer's Perspective*, by Bryant and O'Hallaron (3rd edition, Pearson, 2015, ISBN 978-0134092669). Available at the bookstore and/or from an online retailer.

Description and Assessment of Assignments

- **Availability:** Assignments will be made available on the [Assignments page](#) of the course website.
- **Submissions:** Your assignments must be pushed to your class-provided private repositories on GitHub.
- **Late Submission:** You will be allowed 5 grace days over the semester. You may submit assignment solutions up to 48 hours (2 days) late provided that you still have remaining grace days. Thus, you may only use 2 grace days per assignment. If you do not have grace days, or it is 48 hours past the deadline, no submission will be accepted.
- **Grading and Assessment:** Assignment questions will include a point total that indicates the value of the assignment and each subproblem. For most assignments, a script will be provided to allow you to run the SAME tests we will run for grading, and even show you your final score BEFORE you submit (provided that you follow submission instructions correctly).
- **Solutions:** Solutions to the assignments will NOT be provided. However, you are welcome and encouraged to ask the instructor, other students, or other course staff about any problems you could not solve correctly.

Exams

- Time and Location: There will be **two midterm exams** and **one final**. The midterm exams will be held during the **quiz section** (Fri 4-5:50pm PT). The dates of the exams are shown on the [class schedule](#) but may be moved to a different date in exceptional cases. The exams may also be moved to a different classroom. Always check with the instructor as the listed exam date approaches to confirm the date and time. You are responsible for finding out when and where the exams will be held. Make-up exams will be given if you have a valid excuse (e.g., serious illness or accident, but proof will be required).
- Academic Accommodations: If you have USC approved academic accommodations, please check with your instructor 2 weeks before the exam to determine when and where you will take the exam.
- Exam Style: Exams are designed to not only test your retention of the material, but your ability to apply it to design and analyze new or novel problems. In this way, your mastery and depth of understanding of the course content will be assessed. Some portion of the exam will contain general knowledge questions and be fill-in/multiple choice. However, the majority of points will come from design/analysis problems using skills learned in class. This is where struggling with the assignments and practice problems given in class, discussion, and on our exam preparation page until you truly understand and feel comfortable with each concept will greatly pay off. Students who simply “get the homework done” without reviewing and understanding each facet will often struggle on the exams.
- Preparation: Practice problems and precise topics for the exams will be posted on the class website on the [Exams page](#).

Grading Breakdown

Grades are calculated as follows:

- Assignments 1-5 (various points, equally weighted): 30% of final grade.
- Worst Midterm (100 points): 15% of final grade.
- Best Midterm (100 points): 25% of final grade.
- Final (100 points): 30% of final grade.

Grading Scale

Course final grades will be determined using the following scale. If the overall grade distribution is lower than expected, the scale may be shifted downward; individual tests will not be scaled.

- Score $\geq 94 \Rightarrow A$
- Score $\geq 90 \Rightarrow A-$
- Score $\geq 87 \Rightarrow B+$
- Score $\geq 83 \Rightarrow B$
- Score $\geq 80 \Rightarrow B-$
- Score $\geq 77 \Rightarrow C+$
- Score $\geq 73 \Rightarrow C$
- Score $\geq 70 \Rightarrow C-$
- Score $\geq 67 \Rightarrow D+$
- Score $\geq 63 \Rightarrow D$
- Score $\geq 60 \Rightarrow D-$
- Score below 60 $\Rightarrow F$

Assignment Rubrics

See the section above for relevant assessment procedures for assignments.

Assignment Submission Policy

Submission policies are outlined with the relevant assignment type in the sections above.

Grading Timeline

Assignment grades will generally be posted within 1 weeks of submission. However, most assignments will contain grading scripts that allow you to determine your assignment grade without waiting for our course staff to post the grades.

Exams will generally be graded within 1 week of the exam date.

Additional Policies

None

Course Schedule

A detailed course calendar with readings, assignments, examinations, is provided on the [Schedule page](#) of the class.

IMPORTANT: In addition to in-class contact hours, all courses must also meet a minimum standard for out-of-class time, which accounts for time students spend on assignments, readings, writing, and other academic activities. For each unit of in-class contact time, the university expects two hours of out of class student work per week over a semester.

CSCI 356 Academic Honesty Supplement

The following is a supplement for this course to the university academic honesty guidelines.

- Any code written with help of a course producer, TA, or instructor must be clearly marked with comments about who helped and when. This may not constitute a meaningful portion of your project. The substring **“assistance from”** must appear in the comment acknowledgment near the code on which you received assistance, followed by the name of whom you got assistance from.
- The help you receive from classmates must be limited to **conceptual help**, e.g., how an algorithm you need to implement works as a “big picture” rather than at the level of code and data structures.
- If you receive help from others, whether an instructor, course producer, or fellow student, **follow the “30-minute rule”**:
 - You may discuss high-level ideas and receive hints regarding how to solve portions of the assignments. However, neither party should keep any written record from this discussion. Afterwards, take a 30-minute break and do something unrelated to the course (watching an episode of your favorite show, for example). You may now return to your assignment.
 - When you write a section of code based on help received, add a comment acknowledging the help, including the substring **“assistance from”** as part.
- **You are explicitly prohibited from seeking help outside of course resources for the programming projects.** The following is an exhaustive list of “course resources”:
 - The instructor, TAs, and course producers.
 - Your fellow students. Remember that this means discussing high-level concepts, not sharing code.
 - The textbook (Bryant and O’Hallaron, Computer Systems: A Programmer’s Perspective, 2015).
 - The public portion of the textbook’s website.
 - The course Piazza page for this semester.
 - Course lectures and discussions, along with any notes provided by instructors.
- The previous bullet point means that if you seek information towards solving part of a programming assignment online, find it, and use it, it may result in an academic honesty violation. Do not search for anything related to the assignments online for any reason without explicit permission from the instructor. *This includes “clarification” and “for reference only”.*
- If you use code fragments from the textbook, you must include a comment acknowledging the source, including the substring **“assistance from”** and the textbook citation (textbook name, authors, and page number is sufficient).
- If you are using git, use your private repository so that your code is not available to your fellow students. Course staff will help set you up with a private repository for each assignment.
- **Violations of this code will result in referral to the Student Judicial Affairs Committee. The USC policy is that any violation shall result in an F in the course.**

Academic Conduct

Plagiarism: presenting someone else's ideas as your own, either verbatim or recast in your own words, is a serious academic offense with serious consequences. Please familiarize yourself with the discussion of plagiarism in *SCampus* in Part B, Section 11, "Behavior Violating University Standards" (policy.usc.edu/scampus-part-b). Other forms of academic dishonesty are equally unacceptable. See additional information in *SCampus* and university policies on scientific misconduct, policy.usc.edu/scientific-misconduct

Support Systems

- *Student Counseling Services (SCS)* - (213) 740-7711 (24/7 on call) Free and confidential mental health treatment for students, including short-term psychotherapy, group counseling, stress fitness workshops, and crisis intervention. engemannshc.usc.edu/counseling
- *National Suicide Prevention Lifeline* - (800) 273-8255 Provides free and confidential emotional support to people in suicidal crisis or emotional distress 24 hours a day, 7 days a week. www.suicidepreventionlifeline.org
- *Relationship and Sexual Violence Prevention Services (RSVP)* - (213) 740-4900 (24/7 on call) Free and confidential therapy services, workshops, and training for situations related to gender-based harm. engemannshc.usc.edu/rsvp
- *Sexual Assault Resource Center* For more information about how to get help or help a survivor, rights, reporting options, and additional resources, visit the website: sarc.usc.edu
- *Office of Equity and Diversity (OED)/Title IX Compliance* - (213) 740-5086 Works with faculty, staff, visitors, applicants, and students around issues of protected class. equity.usc.edu
- *Bias Assessment Response and Support* Incidents of bias, hate crimes and microaggressions need to be reported allowing for appropriate investigation and response. studentaffairs.usc.edu/bias-assessment-response-support
- *The Office of Disability Services and Programs* Provides certification for students with disabilities and helps arrange relevant accommodations. dsp.usc.edu
- *Student Support and Advocacy* - (213) 821-4710 Assists students and families in resolving complex issues adversely affecting their success as a student EX: personal, financial, and academic. studentaffairs.usc.edu/ssa
- *Diversity at USC* Information on events, programs and training, the Diversity Task Force (including representatives for each school), chronology, participation, and various resources for students. diversity.usc.edu
- *USC Emergency Information* Provides safety and other updates, including ways in which instruction will be continued if an officially declared emergency makes travel to campus infeasible. emergency.usc.edu
- *USC Department of Public Safety* - UPC: (213) 740-4321 – HSC: (323) 442-1000 (24-hour emergency or to report a crime). Provides overall safety to USC community. dps.usc.edu