

CS351-CUG: Systems Programming

Fall 2023

Course Information

Prof.: Melanie Cornelius

E-mail: melanie.e.cornelius@gmail.com

Office Hours: Appointment Only

Class Hours: M/F 7:00-8:20 AM CST

Course Website: www.mseryn.com/teaching/cs351

TA: Lang Liu

E-mail: lliu94@hawk.iit.edu

Office Hours: Saturday, 6:00 am - 7:00 am CST (Saturday 19:00 pm to 20:00 pm in China)
and Friday, 6:45 am - 7:00 am (before class)

Course Description

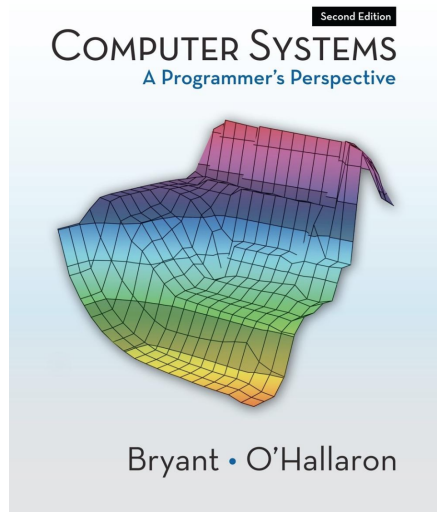
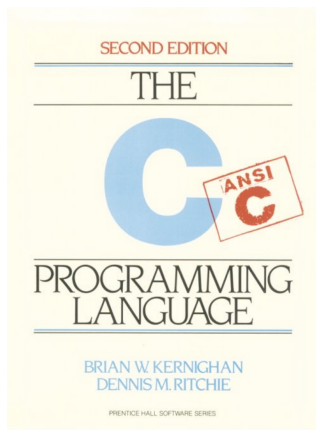
This course focuses on the programming facilities, mechanisms, and abstractions supported by modern operating systems and related low-level libraries and software. In short, we'll be looking at what sorts of things can be done via services provided by operating systems and how to go about doing them in a robust and efficient manner.

To a lesser extent, and only to the degree necessary to enable us to go about our programming tasks effectively, we'll also explore how certain abstractions presented by the operating system work "under-the-hood". For instance, when looking at how to perform low-level I/O it helps to understand some of the relevant data structures used by the operating system to fully appreciate how much time is spent, say, "opening" a file.

Material adapted, with permission, from previous semesters of IIT CS 351 under Prof. Michael Lee, lee@iit.edu, <https://moss.cs.iit.edu/cs351/>

Prerequisites/Corequisites

- "Substantial" programming experience
- Data structures: concepts & implementation



THE LINUX PROGRAMMING INTERFACE

A Linux and UNIX[®] System Programming Handbook

MICHAEL KERRISK



- Basic run-time analysis (big O)
- Knowledge of (any) assembly language
- Data representation (binary, two's comp, f.p. inaccuracy, etc.)
- Von Neumann model (CPU, memory, I/O)
- Stack usage / conventions

Required Materials

- **Required:** Bryant, Randal E., and David O'Hallaron. Computer Systems: A Programmer's Perspective, Third Edition. Prentice Hall, 2015.
- **Recommended:** Kernighan, Brian W., and Dennis M. Ritchie. The C Programming Language, 2nd Edition. Prentice Hall, 1988.
- **Recommended:** Rochkind, Marc J. Advanced UNIX Programming. Addison-Wesley, 2004.

Course Objectives

After successfully completing this class, students should be able to:

1. Define the concept and role of a process in a modern operating system
2. Describe the key abstractions an operating system provides to running processes
3. Inspect and debug running programs at the machine level, and understand how low-level code vulnerabilities such as buffer overflows can be exploited and mitigated.

4. Explain how exceptional control flow works in a modern operating system, and its role in implementing system calls and hardware interrupts.
5. Use the system-level API (system call interface) of a modern operating system for the purposes of process and memory management.
6. Explain the design goals and makeup of the memory hierarchy, as found in modern computer systems.
7. Explain how caches are implemented, and how code can be optimized for cache performance.
8. Explain how the memory hierarchy is used to implement virtual memory, and the tradeoffs of various address translation mechanisms.
9. Explain the goals and responsibilities of a dynamic memory allocator, and implement an explicit dynamic memory allocator.
10. Understand the goals, implementation techniques, and tradeoffs of different garbage collection algorithms.
11. Feel confident delivering a short presentation on systems topics

Course Structure

Lectures

Classes will be held via Zoom on BlackBoard Collaborate. Specific meeting details will be shared, and recordings of each lecture will be posted to Panopto. The schedule for lecture materials will be posted to the course website.

Students are encouraged to attend lectures live. During meetings, polls will be posted where students will be expected to respond - this will be factored into half of the participation grade for each student.

Participation

We will be using Piazza for class discussion. The system is highly catered to getting you help fast and efficiently from classmates, the TA, and myself. Unless a question contains personal information, rather than emailing questions to the teaching staff you should post your questions on Piazza.

Find our class signup link at: <https://piazza.com/iit/fall2023/cs351cug>

During the semester, students will be assigned a topic to discuss with their peers on Piazza. The other half of the participation component of the course will be determined by this assignment. Details will be specified and posted to the course website as well as to BlackBoard.

Labs

There will be 5-7 labs. Most will require you to read and write a substantial amount of code in C and/or assembly. All programming assignments will be submitted via version control system, with specific directions included in the assignment writeup.

Exams

Both the midterm and final will cover all material up to that point. Dates and procedure for remote exam taking will be posted to BlackBoard and the course website.

Grading Policy

The typical Illinois Tech grading scale will be used, potentially with a curve. Any curve will only ever make it easier to obtain a certain letter grade.

Pct	Item
40%	Labwork
25%	Midterm
25%	Final exam (cumulative)
10%	Participation

Late Policy

Assignments are due at 11:59 PM CST on the day they are due.

For late submissions, we will use flexible late days (FLDs). Each student has 6 FLDs this semester. If you need to take a late day, just email the professor *and* TA with your name, SWID, and how many FLDs you are using.

At most, 2 FLDs may be used per assignment.

After the due date (possibly extended with one or two FLDs), a 20% late penalty will be applied per-day.

Of course, if you have any emergency or outstanding reason for a late submission, please reach out to the professor. You don't need to use a FLD in such cases.

Academic Integrity

You are welcome to discuss assignments with classmates, but all final work must be your own. Academic dishonesty of any kind may result in a 0 on the assignment, a reduction in final grade, and/or referral to the Dean.

The IIT code of Academic Honesty may be found in the undergraduate handbook.

Accommodations for Disabilities

Reasonable accommodations will be made for students with documented disabilities. In order to receive accommodations, students must obtain a letter of accommodation from the Center for Disability Resources. The Center for Disability Resources (CDR) is located in Life Sciences Room 218, telephone 312 567.5744 or disabilities@iit.edu.

Sexual Harassment and Discrimination Information

Illinois Tech prohibits all sexual harassment, sexual misconduct, and gender discrimination by any member of our community. This includes harassment among students, staff, or faculty. Sexual harassment of a student by a faculty member or sexual harassment of an employee by a supervisor is particularly serious. Such conduct may easily create an intimidating, hostile, or offensive environment.

Illinois Tech encourages anyone experiencing sexual harassment or sexual misconduct to speak with the Office of Title IX Compliance for information on support options and the resolution process.

You can report sexual harassment electronically at iit.edu/incidentreport, which may be completed anonymously. You may additionally report by contacting the Title IX Coordinator, Virginia Foster at foster@iit.edu or the Deputy Title IX Coordinator at eespeland@iit.edu.

For confidential support, you may reach Illinois Tech's Confidential Advisor at (773) 907-1062. You can also contact a licensed practitioner in Illinois Tech's Student Health and Wellness Center at student.health@iit.edu or (312)567-7550

For a comprehensive list of resources regarding counseling services, medical assistance, legal assistance and visa and immigration services, you can visit the Office of Title IX Compliance website at <https://www.iit.edu/title-ix/resources>.