

CS 551System Programming Fall 2024

Professor Ken Mandelberg

Office Hours (Zoom Tu/Th 4:5 meeting id 946 5385 3632)

Email km@emory.edu

I check email many times a day, and try to reply quickly

Teaching Assistants: Office Hours and email TBA

Final Exam : 12/15 11:30-2:00

NOTE: I will not be using Canvas in this course. As a system course I want to emphasize using the Linux environment. Materials will be on the labsystems under the ID cs551001. A limited number of documents (including this one) will available by the web at <http://cs.emory.edu/~cs551001>, most files will be in the cs551001 share directory.

Course Prerequisites

This course assumes a basic background in computer architecture, data structures, and a working knowledge of C programming. The assignments are difficult C programs.

Course Overview

Computer programming can be divided into Application Programming and Systems Programming. Until now you have mainly done the former. Application Programming is mainly concerned with algorithmic business logic, but is only casually aware of the Operating System on which the program runs. Systems Programs take advantage of the Operating Systems interface to gain access to the hardware and software resources that the Operating System Provides to increase performance and provide information.

We will be studying the system interface know as POSIX (portable operating system interface) which is most completely implemented on Open Systems, most notably Linux. In this course besides writing programs, you will be asked to become proficient with the Linux command line tools, including the shell and gdb.

Required Reading

The textbook is Advanced Unix Programming by Rochkind. While the lectures will cover many examples and details, the text is a necessary supplement of further documentation and examples.

Technology Requisites

Linux programming and the command line environment are the essentials parts of the course. We provide access to the Math/CS computer lab of Linux workstations running Ubuntu. All programs will be tested there. The computer lab can be accessed both physically and remotely, TA office hours will be in the in lab. In addition you might find it convenient to install Ubuntu on your computer, for example via Virtualbox.

Course Topics (time permitting)

C programming and the Linux Command Line

We will revisit advanced topic in C programming by comparison to Java, as well as introduce how the command line (shell) works, shell scripts, and gdb debugging.

The Filesystem

There are many filesystems available with different histories and requirements. We will study the most basic one (UFS) in the Open System space that is the basis of the newer technologies. We also will study how the filesystem is a unifying concept between hardware and software.

File Programming

We study the programming interfaces to access files in more efficient ways and how to deal with metadata programmatically.

Computer Processes

We study how computer programs execute becoming computer processes, and how the operating system manages them. We study how processes can create new processes.

Interprocess Communication

Now that we know how processes can create additional processes, we need to study how a large task can benefit by using multiple processes. This involves communicating between processes. The techniques for this are a major portion of the course.

Asynchronous Programming

Systems Programs in particular are often event driven. We study how signals work.

Thread Programming

While much of the course is on multiprocess computing, we will also study thread programming for leveraging multiple cpus in a shared address space.

Network Programming

We will study client server computing on a network including heterogeneous architectures..

Container and VM Technology

Separating the native environment from the application environment

Course Logistics

There will be 5 or 6 computer programs counting 60% of the grade. A midterm and quizzes will count 15% and a final exam 25%.

The computer programs are essentially take home exams. What you hand in has to be your own work. You can't copy from the internet, from another student (past or present), or a tutor. That doesn't mean that discussions on strategy or debugging techniques are bad, in fact they are encouraged.

However, in this course there is the additional restriction that no assignment code can be written or derived from code produced by any of the AI technologies (ChatGPT, Gemini, Copilot, etc). This means the code you submit MUST have the comment statement:

THIS CODE IS MY OWN WORK, IT WAS WRITTEN WITHOUT CONSULTING A TUTOR,
CODE WRITTEN BY OTHER STUDENTS, OR CODE DERIVED FROM AN AI TOOL- YOUR
NAME