2020FA - COMPUTER ARCHITECTURE 01:198:211:05-08

Lecture: Monday noon-1:20pm

Thursday noon-1:20pm
Synchronous Remote Lectures via Webex

Instructor: Prof. Santosh Nagarakatte (http://www.cs.rutgers.edu/~santosh.nagarakatte/)

LMS: Canvas - https://rutgers.instructure.com/courses/65852

Contact: santosh.nagarakatte@cs.rutgers.edu
Office Hours: CORE 328,
Thursday 2pm-3pm or by appointment

Course Description

This course covers materials to illustrate how a high level program is converted to binary and is eventually executed by the hardware. We will cover a brief introduction to programming in C, how data and control represented in C is converted to binary and how the binary code is executed by the processor. This will involve programming in C, understanding and debugging X86 assembly, interfaces between the hardware and the software components, organization of hardware and how the various components interact.

1. Von Neumann architecture, hardware trends, performance, energy, cost metrics
2. Introduction to C programming
3. Assembly language techniques and instruction-set design
4. Boolean algebra, digital logic
5. The memory hierarchy
6. Performance
7. Input/Output concepts

Prerequisites

CS 112 is a pre-requisite for this course.

Textbooks


- Students can probably also use the second edition of the book.

- Note that while this textbook is recommended, we will use portions of it for various lectures.

Recommended for C:

Modern C by Jens Gustedt. The online version is available for free at [https://modernc.gforge.inria.fr/](https://modernc.gforge.inria.fr/)

The Hardware/Software Course by Luis Ceze and Gaetano Borriello from University of Washington is available on Youtube and is useful: [https://www.youtube.com/playlist?list=PL0oekSefhQVJdk0hSRu6sZ2teWM740NtL](https://www.youtube.com/playlist?list=PL0oekSefhQVJdk0hSRu6sZ2teWM740NtL)

**Grading**

Assignments - 70%

- PA1 - 5%
- PA2 - 15%
- PA3 - 10%
- PA4 - 10%
- PA5 - 15%
- PA6 - 15%

Class + Recitation participation - 4%

Timed Quizzes - 26%

- 13 timed quizzes each worth 2% of the grade

**Programming Assignment Policies**

There will be numerous programming assignments to improve your understanding of the subject.

Programming assignments will be handed in via Canvas. We will provide instructions for packaging and handing in your assignments. You must follow these instructions exactly. If we cannot compile or run your programs, you will lose a significant portion of points.

We will not accept late assignments. Programming assignments must be handed in by the specified time/date on Canvas.
If you don't already have an account on the Instructional Laboratory cluster go to

https://www.cs.rutgers.edu/resources/instructional-lab

and create one. All CS students have remote access to the iLab machines. See iLab machines at

https://resources.cs.rutgers.edu/docs/instructional-lab/


Some other manuals that will be useful: http://www.gnu.org/software/emacs/manual

(Emacs) http://gcc.gnu.org/onlinedocs/

(Make) http://www.gnu.org/software/make/manual/

(Gdb) http://www.gnu.org/software/gdb/documentation

Rules for Collaboration

You are free, even encouraged, to talk to your fellow classmates about your assignments. However, you cannot copy from one another in the assignments. If in doubt, follow Gilligan's Island rule. That is, after a joint discussion of an assignment, each student should discard all notes, go do something mind-numbing for an hour (like watching a couple of episodes of Gilligan's Island), and then recreate the solutions absolutely individually. The idea is to ensure that you fully understand and provide your own solutions, rather than blindly incorporate the solutions from the group discussion. We will use sophisticated plagiarism detection software and report any violation to office of student conduct.

You must also follow the Department's Academic Integrity Policy:
https://www.cs.rutgers.edu/academic-integrity/introduction

Finally, copying from the web or from GitHub is also cheating. Posting assignments, exams, and quizzes on Chegg is a violation of Rutgers and CS Academic Integrity Policy.

Syllabus

September 3, 2020 -- Lecture 1: Introduction and Hardware Trends

Required Reading: Chapter 1 of the text book
September 8, 2020 (Modern C book)-- Lecture 2: C Programming

September 10, 2020 (Modern C book)-- Lecture 3: C Programming

September 14, 2020 (Modern C book)-- Lecture 4: C Programming

September 17, 2020 (Modern C book) -- Lecture 5: C Programming

September 21, 2020 (Modern C book) - Lecture 6: C Programming

September 24, 2020 (Modern C book) : Lecture 7: C Programming

September 28, 2020: Lecture 8 : Data Representation

**Reading material:** Chapter 2:Introduction, 2.1 (skip 2.1.7), 2.2, 2.4-2.4.3

October 1, 2020: Lecture 9: Data Representation

**Reading material:** Chapter 2: 2.1, 2.2, 2.3, 2.4-2.4.3, 2.5

October 5, 2020: Lecture 10 (Chapter 3: Everything but 3.14) -- Assembly Language Programming

October 8, 2020: Lecture 11 -- Assembly Language Programming

October 12, 2020: Lecture 12 -- Assembly Language Programming

October 15, 2020: Lecture 13 -- Assembly Language Programming

October 19, 2020: Lecture 14 - Assembly Language Programming

October 22, 2020: Lecture 15- Assembly Language Programming

October 26, 2020: Lecture 16- Assembly Language Programming

October 29, 2020 : Timed Quiz/Exam

November 2, 2020: Lecture 17 - Assembly

November 5, 2020: Lecture 18 - Caches

November 9, 2020: Lecture 19 -- Caches

November 12, 2020: Lecture 20 - Digital Logic

November 16, 2020 - Lecture 21 - Digital Logic

November 19, 2020 - Lecture 22 -- Digital Logic
November 23, 2020 - Lecture 23 -- Digital Logic

November 30, 2020 - Lecture 24 - Digital Logic

December 3, 2020 - Lecture 25 -- Digital Logic

December 7, 2020 - Lecture 26 -- Digital Logic

December 10, 2020 - Lecture 27 -- Digital Logic