CS 211: Computer Architecture

Spring 2019

Syllabus (subject to change)

Overview

This is a first course on computer organization and architecture. We will cover an introduction to processor design, focusing on their fundamental logic organization. This will involve understanding its various parts, how they interact. Furthermore, students will be exposed to processor native languages, and system software concepts. For example, the following topics will be covered in the course:

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

Staff

Instructor:

- Jeff Ames (jca105@cs.rutgers.edu)

TAs:

- Abdall Ahmed (aa1494@scarletmail.rutgers.edu)
- Abu Awal Shoeb (as2352@scarletmail.rutgers.edu)
- Bo Zhang (bz186@rutgers.edu)
- Zhou Qin (zq58@cs.rutgers.edu)

Prerequisites

CS 112 is a pre-requisite for this course.

Textbooks

- Required: R.E.Bryant and D.R.O'Hallaron, Computer Systems: A Programmer's Perspective, Prentice Hall
- Recommended: D.A.Patterson and J.L.Hennessy, Computer Organization and Design: The Hardware/Software Interface, Morgan Kauffman

  Note that while this textbook is recommended, we will use portions of it for various lectures.
- Recommended for C: B.W.Kernighan and D.M.Ritchie, The C Programming Language, Prentice Hall

  Note that while this is the classic reference for C, there are a number of online references available as well. For example, here is a free C online reference http://publications.gbdirect.co.uk/c_book/

  While you are welcome to use these resources, note that we have not seen them ourselves and cannot attest to the usefulness of these references.

Grading

Homework assignments	40%
Midterm exam	30%
Final exam	30%

All grading disputes must be raised within one week of grades being returned. After this, grades are considered to be final.

Exams

- Midterm: March 14
- Final: May 10, 12:00 pm – 3:00 pm

Programming Assignment Policies

There will be multiple programming assignments.

Programming assignments will be handed in via Sakai. 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 Sakai.

If you don't already have an account on the Instructional Laboratory cluster go to https://www.cs.rutgers.edu/resources/getting-started-with-technical-resources-at-the-department-of-computer-science and create one. iLab machines are located in Hill 248, 250, 252, and 254. All registered students have access to these rooms.

In addition, there are a number of references to help you understand how to use the iLab systems. Please look at: https://www.cs.rutgers.edu/resources/beginners-info.

You will be using a Unix environment for the programming assignments. For a primer on using the Bash environment, please refer to https://tldp.org/LDP/Bash-Beginners-Guide/html/index.html For an assortment of GNU/Linux command line tools, refer to this: https://www.tldp.org/LDP/GNU-Linux-Tools-Summary/GNU-Linux-Tools-Summary.pdf

Some other manuals that will be useful:

- (GCC) https://gcc.gnu.org/onlinedocs/
- (Emacs) https://www.gnu.org/software/emacs/manual
- (Make) https://www.gnu.org/software/make/manual/
- (Gdb) https://www.gnu.org/software/gdb/documentation

Homework Assignment Policies

Late assignments will not be accepted.

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.

You must also follow the Department's Academic Integrity Policy:
https://www.cs.rutgers.edu/policies/academicintegrity/

Finally, copying from the web is also cheating.

Tentative schedule

Lecture	Date	Topic	Reading
1	1/22	Introduction and Hardware Trends	Ch. 1
2	1/24	C Programming	K&R
3	1/29	C Programming	K&R
4	1/31	C Programming	K&R
5	2/5	C Programming	K&R
6	2/7	C Programming	K&R
7	2/12	C Programming	K&R
8	2/14	Data Representation	Ch. 2: Introduction, 2.1 – 2.4
9	2/19	Computer Arithmetic	Ch. 2: 2.3 – 2.5
10	2/21	Assembly	Ch. 3: Introduction, 3.1 – 3.5
11	2/26	Assembly	Ch. 3: Introduction, 3.1 – 3.5
12	2/28	Assembly	Ch. 3
13	3/5	Assembly	
14	3/7	Assembly	
15	3/12	Assembly	
16	3/14	*midterm*	
	3/19	(Spring break)	
	3/21	(Spring break)	
17	3/26	Assembly	
18	3/28	Assembly	
19	4/2	Digital Logic	Ch. 4: 4.2
20	4/4	Digital Logic	Ch. 4: 4.1, 4.3
21	4/9	Digital Logic	
22	4/11	Digital Logic	
23	4/16	Digital Logic	
24	4/18	Digital Logic	
25	4/23	Digital Logic	
26	4/25	Digital Logic	
27	4/30	Caches	
28	5/2	Caches	

Acknowledgments

This course borrows from CS 211 taught by Prof. Abhishek Bhattacharjee, Prof. Santosh Nagarakatte, Prof. Badri Nath, and Prof. Thu Nguyen.