COURSE:

Systems 1: Computer Organization and Architecture

DESCRIPTION:

The first course in the Systems series helps students understand the basic operation of computing hardware, how it works, and how it interfaces to software. Upon completing this course, students should have a high-level understanding of the role played by compilers, assemblers, instruction sets, and hardware. Students will also learn system-level programming and apply their knowledge of computer architectures to programming for performance.

INSTRUCTOR:

Dr. Robert F. Dickerson
GDC 5.318
rfd@cs.utexas.edu

Teaching Assistants:

- Lijia Liu (lijialiu@cs.utexas.edu)
- Kai Wang (redudie@gmail.com)
- Keshav Kini (krkini@utexas.edu)
- Yi-Chao Chen (yichao@cs.utexas.edu)

LECTURES

Lectures are held in Gearing Hall (GEA) 105 (Links to an external site.). You may attend either lecture that works best for you. The course runs from Jan 21 to May 6 on Mondays and Wednesdays.

2:00 - 4:00 PM
4:00 - 6:00 PM

The lecture content should follow this schedule. However, things might change

Week 1: Introduction + Intro to C
Chapter 1

Week 2: Bits and Bytes
Chapter 2.1 - 2.3

Week 3: Floating point + Machine Programming 1
Chapter 2.4 + Chapter 4.1

Week 4: Machine Programming 2 + Control 1
Chapter 3.1 - 3.6

Week 5: Control 2 + Procedures 1
Chapter 3.7

Week 6: Procedures 2 + Midterm 1 Review

Week 7: Midterm 1 + Data 1

Week 8: Data 2 + Datapath 1
Chapter 3.8 - 3.9 + Chapter 4.2

Spring break

Week 9: Datapath 2 + Midterm Review
Chapter 4.3

Week 10: Midterm 2 + Guest Lecture

Week 11: Pipelining 1 + Pipelining 2
Chapter 4.4 - 4.5

Week 12: Pipelining 3 + Pipelining 4

Week 13: Memory Hierarchy + Caching
Chapter 6.1 - 6.3

Week 14: Review + Midterm 3

Week 15: Caching 2 + Review
Chapter 6.4 - 6.6

LABS

The following classrooms are used for weekly lab meetings with your TA. Please attend the lab meeting that your section is registered for because this is where the TAs will hand back exams and other material.

(51775) PAR 301 - 10:00 AM to 12 PM - Kai
(51780) PAR 201 - 12:00 PM to 2 PM - Lijia
(51785) GDC 5.302 - 2:00 PM to 4 PM - Lijia
(51790) WAG 214 - 9:00 AM to 11 AM - Keshav
(51795) BUR 224 - 11:00 AM to 1 PM - Keshav
(51800) WEL 2.256 - 1:00 PM to 3:00 PM - Kai

The lab session is normally divided into two parts, the first hour and the second hour. The first hour is lead by the TA to introduce the biweekly lab that has been assigned and any lecture that he or she has prepared. The second hour is used to discuss problem sets that you would like to work on (like office hours).

TEXTBOOK

Computer Systems: A Programmer's Perspective, Second Edition (Links to an external site.) Randal E. Bryant and David O'Hallaron, ISBN 0-13-610804-0, 2011. (Used for both Systems 1 and 2)
Programming from the Ground Up (Links to an external site.) Jonathan Bartlett

The C Programming Language (Links to an external site.) Brian Kernighan and Dennis Ritchie

STUDENT EVALUATION

Your performance in this class will be evaluated through participation, 4 homework assignments, 6 laboratory assignments, 3 midterm exams, and a final exam. It is best to assume there will be no "curving" these scores. The weights of each of these components is listed below:

- Exams: 50%
- Laboratory Assignments: 40%
- Homeworks: 5%
- Class participation: 5%

LAB ASSIGNMENTS

There are 7 laboratory assignments, scheduled roughly every 2 weeks. Laboratory assignments are due at 9 AM on the Friday that it is due. Programming components of the laboratory assignments must be tarballed and submitted electronically through Canvas. You may not email your assignments to the teaching staff since Canvas helps us organize submissions better. Programs will be graded on correctness, readability, style, documentation, and performance. You must work alone on these assignments, without collaboration.

- Lab 0 - Linked List in C - Using Linux, programming and compiling in C
- Lab 1 - Data Lab - using bit arithmetic to solve problems
- Lab 2 - Y86 programming - write programs in Assembly
- Lab 3 - Defusing a Bomb - reverse engineering to diffuse Dr. Evil's bomb
- Lab 4 - Buffer bomb - stop Dr. Evil from reaching into bad memory addresses
- Lab 5 - Extending the Machine - Add new instructions to a processor
- Lab 6 - Improving performance - improve the performance of a pipelined processor

EXAMS

There are 4 exams during the term, scheduled roughly every month. The 3 midterms during the course term evaluate your understanding of new material. The final exam is comprehensive and tests your understanding of the totality of the course. The exams are closed-book and closed-notes. No electronic devices are necessary to solve the problems, and reference sheets will be printed out to assist your memory over ISA's and other material. When we hand out the exams, we must unfortunately have them handed back in order to prevent exams floating around on the black market.

You should in most cases take your midterm during the lecture session you are registered for, however in some cases, you may take the exam in the lecture you have been attending regularly. If this relaxed policy becomes a problem, I might restrict this in future exams. The current midterm dates are the following. Assuming we keep up with schedule, they will be:

Midterm 1 (C Programming + Bits and Bytes) , March 2
Midterm 2 (Assembly Programming), March 30
Midterm 3 (Datapath + Pipelining), April 27
Final Exam (Comprehensive + Caching), TBA

HOMEWORKS

There are only 4 homework assignments which are challenging and helps solidify your knowledge. Most problems come from the book, with other supplementary problems added. You must prepare your solutions as a PDF document, and upload it to Canvas on the Thursday in which it is due. You might want to use LaTeX for typesetting your document. You may work with friends on this assignment.

PARTICIPATION

Attending every lecture is important for understanding the material. In order to make the lecture more exciting, I will be asking questions intermittently when presenting new information. Since we are in a large classroom, we will be using the iClickr device to tally your responses to the problem. The accuracy of the responses may or may not have an impact on your participation score, however participation is important for me to know if I should slow down and explain how a problem should be answered.

You should actively participate in Piazza. It's not only for posting questions you have, but also posting answers to your classmates' questions. If you discover something interesting, such as a vulnerability in the Linux kernel or something that's relevant, please share it. Your participation on Piazza can be tracked with stats about how often you respond to questions and it could have an impact positively on your participation score.

HOW TO DO WELL IN THIS COURSE:

- Show up punctually and consistently for the lectures and laboratory sections.
- Find friends to study with. Work on practice problems together, or better yet, pick a problem and then teach each other how to solve it.
- Check Canvas frequently for latest course announcements and information
- Submit questions and respond to other student's questions on Piazza. Engage the material and ask things or share new things you learned on Piazza as well.
- Get familiar with the GNU/Linux environment. Get a department account, and start learning the Unix operating system sufficiently well to meet the demands of the course (gcc, gdb, emacs, vi, make, etc). Install Linux in a virtual machine or on a partition on your personal computer. You can find many GNU/Linux guides online.
- Do the posted readings on time. Although the book is good, it is quite dense in material and a bit dry. Pacing your readings will help with retention. Complete all the practice problems when you read, and check your answer in the back on the chapter.
- Doing well in this course requires a strong dedication to learning the material and may require a substantial time commitment to complete the programming assignments. Start early, and seek help!