CS 4400: Computer Systems

Fall Semester 2014

Instructor:

Paul Rosen (prosen@sci.utah.edu)
Office: WEB 3809
Office Hours: M 3:00-4:00pm or by apt.

Teaching Assistants:

Scotty Bauer (sbauer@eng.utah.edu)
Sarah Spall (sspall10@gmail.com)
Chitradeep Duttaroy (rahduro@gmail.com)
Saurav Singh (u0877003@utah.edu)
Yihao Jiang (yihao.jiang@utah.edu)

Class Location: WEB L101

Class Time: M/W 1:25 – 2:45 p.m.

Lab Location: CADE Lab

Course Information

The objective of this course is to help students bridge the gap between high-level programming and actual computer systems: processors, the memory hierarchy, operating systems, compilers, linkers, assemblers, networks, and more.

Our basic goal is to understand how a computer works, so that as programmers, we can make it work efficiently. Thus, this course is an introduction to computer systems from a programmer's point of view.

The official prerequisite for this course is CS 3810 (Computer Organization). *It is strongly recommended that students complete CS 3505 (Software Practice II) before taking this class.*

Course Materials and Getting Help

The required course text is *Computer Systems: A Programmer's Perspective* by Randal E. Bryant and David R. O'Hallaron, 2nd edition, Prentice-Hall, 2010 (ISBN: 0-13-610804-0). An e-version is available at http://www.coursesmart.com/0132130661/?a=1773944.

A highly recommended book is *The C Programming Language* by Brian W. Kernighan and Dennis M. Ritchie, 2nd edition, Prentice-Hall, 1988 (ISBN: 978-0131103627).

Canvas will contain all of the resources required for the class (references, assignments, grades, etc.). Notes will usually be posted after but occasionally before class sessions.

- **Canvas: https://utah.instructure.com/**

See the TA calendar for the consulting schedule of the course TAs. During consultation, use the TA Queue Signup (also on the class website) to alert the TA on duty that you have a question.

- **TA Calendar: http://bit.ly/1z6Cmk5**
- **TA Queue: http://nine.eng.utah.edu/schools/uofu/prosen/CS4400/login**

If you have a question about the class or an assignment, please send it to…

- **CS 4400 Instructor Mailing List: teach-cs4400@list.eng.utah.edu**

The course staff will use the class mailing list cs4400@list.eng.utah.edu to send urgent messages to everyone in the class, such as corrections to assignments or changes in due dates. Students are not able to send mail to the class list. Students **MUST** subscribe to the class mailing list ASAP.

- **Mailing list signup: https://sympa.eng.utah.edu/sympa/info/cs4400**

Course Work

Lab Work

Because this is a 4-unit course, there is a significant amount of lab work in the form of programming assignments.

The lab work makes heavy use of C, Linux, and the Intel IA32 (x86) architecture. Students not currently fluent in any of these three topics should not panic, as this course will cover them in more detail throughout the semester. However, there is an assumption that students have some familiarity with C or C++. Students should be prepared to learn some of the C programming language on their own, for which the Kernighan and Ritchie reference text will be very useful.

All lab work must use an x86 processor (either AMD or Intel) that has a Unix OS. *Code must be in ANSI standard C---nothing else will work.* Unless otherwise noted, grading of assignments will be done using CADE Lab 1 machines. Students who choose to develop their code on any other machine are strongly encouraged to run their programs on a CADE Lab 1 machine before turning it in. *There will be no credit for programs that do not compile and run on a CADE Lab 1 machine*, even if they run somewhere else.

CADE Lab 1 machines are numbered 1-48 and have names `lab1-x.eng.utah.edu`, where `x` is the machine number. For more information on the CADE lab and how to remotely log into these machines, see http://www.cade.utah.edu.

Labs submitted late will receive a ZERO. Every student is allocated one (1) '*late pass*', which they may use on any lab. A late pass gives the student one (1) extra week to turn in the lab without penalty. Other exceptions to the late policy will only be made on a case-by-case basis for legitimate cause (unexpected visits to the hospital, etc.). Evidence of the cause is required.

Exams

Two midterm exams will take place in class **OCTOBER 1** and **NOVEMBER 12**. The final exam is cumulative and will take place **DECEMBER 16**, 1-3p in WEB L101.

Problem Sets

Every week, a number of problems from the textbook will be assigned. Your solutions will due by 1:25p on the following Wednesday via canvas. Unless explicitly noted otherwise, **all submissions must be in PDF format**. The solutions must be clear and easy to ready, but are not required to be typed. **Problem sets are not accepted late**. Solutions to the problems will be discussed in class and/or posted to canvas.

Grades

The final course grade will be based on evenly-weighted lab-work assignments (50% total), two evenly-weighted midterm examinations (20% total), a final examination (15%), and weekly problem sets (15%). At the instructor's discretion, there may be an upward curve (but no downward curve).

Scale for assigning letter grades:

		89-87	B+	79-77	C+	69-67	D+		
100-93	A	86-83	B	76-73	C	66-63	D	59-0	E
92-90	A-	82-80	B-	72-70	C-	62-60	D-		

Students who wish to appeal a grade on an assignment or exam, must do so within one week of receiving the grade.

Behavior in the classroom

All students are expected to maintain professional behavior in the classroom setting, according to the University of Utah Student Code, which is posted at http://www.regulations.utah.edu/academics/6-400.html. Students should read the Code carefully and know that they are responsible for the content. According to Faculty Rules and Regulations, it is the faculty responsibility to enforce responsible classroom behaviors, beginning with verbal warnings and progressing to dismissal from class and a failing grade. Students have the right to appeal such action to the Student Behavior Committee.

Academic Honesty

I am a strong believer in collaborative problem solving. Nevertheless, you must do your own work. It is perfectly acceptable to discuss the solution to any homework problem with your classmates, but YOU MUST COMPLETE THE ASSIGNMENT YOURSELF!

Cheating is: sharing code or other files either by copying, retyping, looking at, or supplying a copy of a file.

Cheating is not: discussing concepts, answering questions about concepts or clarifying ambiguities, helping someone understand how to use the computer systems or basic tools (compiler, debugger, etc.), or helping with high-level design issues or general debugging.

Any student found cheating will be reported to the dean's office and fail the entire course.

College of Engineering Guidelines

For information on withdrawing from courses, appealing grades, and more, please see College Guidelines at http://www.coe.utah.edu/academics.

A.D.A. Statement

Reasonable accommodation is provided to students with known physical, sensory, cognitive, systemic, learning, & psychiatric disabilities. If you will need accommodations in the class, please contact the Center for Disability Services (http://disability.utah.edu/) - 162 Olpin Union Building, (801) 581-5020.

Syllabus

The following are the key topics planned for study, the approximate number of lectures devoted to each, and the corresponding chapters in the course text.

Getting Started (1 lecture) Chapter 1
- Administrative details
- Overview of computer systems

Representing Information (8 lectures) Chapters 2-3
- Bits and bytes (information storage)
- Integers (representation, arithmetic)
- Floating point (representation)
- x86 machine-level code (accessing information, operations)
- Control flow (jumps, branches)
- Procedures (run-time stack, recursion)
- Data (arrays, pointers, structures, alignment)

Optimizing Code (2 lectures) Chapter 5
- Optimizing compilers
- Loops
- Branch prediction
- Memory performance

The Memory Hierarchy (2 lectures) Chapter 6
- Different kinds of memory
- The principle of locality
- Cache memory

Running Programs on a System (8 lectures) Chapters 7-10
- Linking
- Exceptions
- Processes
- Signals
- Virtual memory and address translation
- Dynamic memory allocation

Interaction Among Programs (3 lectures) Chapters 12-13
- Network programming
- Concurrent programming