Introduction to Computer Systems

Syllabus

Web Page

http://www.cs.northwestern.edu/~pdinda/icsclass

Instructor

Peter A. Dinda
1890 Maple Avenue, Room 338
847-467-7859
pdinda@cs.northwestern.edu
Office hours: Thursdays, 2-4pm

Teaching assistants

Bin Lin
1890 Maple Avenue, Room 240
847-491-7159
b-lin@cs.northwestern.edu
Office hours: Tuesdays and Thursdays, 10:30-noon.

Bob Adolf
r-adolf@northwestern.edu
Office hours: appointment

Location and Time

Lecture: 1890 Maple Avenue, Small Classroom (342) TR, 5-6:30pm
Recitation: 1890 Maple Avenue, Small Classroom (342) W, 5-6:30pm.

Prerequisites

Required	CS 211 or equivalent
Required	Experience with C or C++
Useful	CS 311 or equivalent

Starting in Fall, 2005, CS 213 is a required CS course. It is also now
a prerequisite for operating systems (CS 343). It is becoming a prerequisite
for *all* of the systems courses (see http://nsrg.cs.northwestern.edu for
a current list)

Textbook

Randal E. Bryant and David R. O'Hallaron*, Introduction to Computer Systems: A
Programmer's Perspective,* Prentice Hall, 2003, (ISBN 0-13-034074-X)
(Required - Textbook)

- Details on http://csapp.cs.cmu.edu

Brian W. Kernighan and Dennis M. Ritchie, *The C Programming Language, Second Edition*, Prentice Hall, 1988 (ISBN 0-131-10370-9) (Recommended)
- Definitive book on C

Richard Stevens, *Advanced Programming in the Unix Environment*, Addison-Wesley, 1992 (ISBN 0-201-56317-7) (Recommended)
- Describes how to think like a Unix systems programmer

Objectives, framework, philosophy, and caveats

This course has four purposes. First, you will learn about the hierarchy of abstractions and implementations that comprise a modern computer system. This will provide a conceptual framework that you can then flesh out with courses such as compilers, operating systems, networks, and others. The second purpose is to demystify the machine and the tools that we use to program it. This includes telling you the little details that students usually have to learn by osmosis. In combination, these two purposes will give you the background to understand many different computer systems. The third purpose is to bring you up to speed in doing systems programming in a low-level language in the Unix environment. The final purpose is to prepare you for upper-level courses in systems.

This is a learn-by-doing kind of class. You will write pieces of code, compile them, debug them, disassemble them, measure their performance, optimize them, etc.

This course is ideally taken after CS 211 early in your academic career.

Resources

You'll be able to do the programming assignments on any modern Linux machine. Consult the course web page for how to get an account on the TLAB machines. It should be possible to do some of the assignments using the Cygwin environment on Windows or in Mac OS X, but they will be graded in a Linux environment, so be sure your code works on the TLAB machines.

Labs

There will be four programming labs. Their goal is to make you apply the concepts you've learned and to gain familiarity with Unix tools that can help you apply them. Labs should be done in groups of two.

Homework

Four problem sets will be assigned. Their goal is to help you improve your understanding of the material. Homework should be done alone.

Exams

There will be a midterm exam and a final exam. The final exam will not be cumulative.

Grading

        10 %   Homeworks (2.5% per homework)
        50 %   Programming labs (12.5% per lab)
        20 %   Midterm (covers first half of the course)
        20 %   Final (covers second half of the course)

Final grades will be computed in the following way.   A final score from 0 to 100 will be computed as a weighted sum of the homeworks, programming labs, and the exams. Scores greater than 90 or greater than 90th percentile will be assigned As, scores greater than 80 or greater than 80th percentile will be assigned Bs, scores greater than 70 or greater than 70th percentile will be assigned Cs, scores greater than 60 or greater than 60th percentile will be assigned Ds, and the remainder will be assigned Fs.   Notice that this means that if everyone works hard and gets >90, everyone gets an A. Please choose wisely where you use your time.

Peter ultimately assigns all grades. If you have a problem with a grade, you are welcome to bring it up with either Peter or Bin, but only Peter is empowered to change grades.

Late Policy

For each calendar day after the due date for a homework or a lab, 10% is lost. After 1 day, the maximum score is 90%, after 2 days, 80%, etc, for a maximum of 10 days.

Cheating

Since cheaters are mostly hurting themselves, we do not have the time or energy to hunt them down. We much prefer that you act collegially and help each other to learn the material and to solve development problems than to have you live in fear of our wrath and not talk to each other. Nonetheless, if we detect blatant cheating, we will deal with the cheaters as per Northwestern guidelines.

Schedule

Lecture	Date	Topics	Readings	Homework/Labs
Note that Tuesday, March 29 has the MONDAY schedule. We will meet for the first time on Thursday, March				
1	3/31 R	Mechanics, Introduction, overview of abstractions using web request-response *Note: Instructor Out of Town. Class will meet with substitute teacher*	Chapter 1	Data lab out

2	4/5 T	Physics, transistors, photolithography, Moore's Law, bits, bytes, and logic	2, 2.1, handout	HW 1 out,
3	4/7 R	Integers and integer math	2.2-2.3	
4	4/12 T	Floating point	2.4-2.5	
5	4/14 R	The Machine Model – instruction set architecture, microarchitecture, and basic instructions	3, 3.1-3.5, 5.7	HW 1 in, HW 2 out
6	4/19 T	Control flow *Note: Instructor Out Of Town. Class may be cancelled.*	3.6	Data lab in Bomb lab out
7	4/21 R	Procedures	3.7	
8	4/26 T	Data	3.8-3.11	
9	4/28 R	Advanced machine code	3.12-3.16	HW 2 in, HW 3 out
Midterm Exam: Monday, 5/2, 6-8pm, CS Small Classroom.				
10	5/3 T	Memory and cache	6, 6.1-6.4	
11	5/5 R	Cache performance	6.5-6.7	Bomb lab in, Exploit lab out
12	5/10 T	Linking	Chapter 7	
13	5/12 R	Exceptional control flow	8,8.1-8.4	
14	5/17 T	Exceptional control flow	8.5-8.8	HW 3 in
15	5/19 R	Virtual memory Memory system	10, 10.1-10.8	Malloc lab out, Exploit lab in
16	5/24 T	Memory allocation	10.9-10.13	HW 4 out
17	5/26 R	Input and Output	Chapter 11	
18	5/31 T	Network programming	Chapter 12 handout	
19	6/2 R	Concurrency, Distributed Systems and Wrap-up	Chapter 13 handouts	Malloc lab in HW 4 in
Finals week – Exam is sometime during finals week				

Note that in the latter part of the course, we will cover Chapters 11-13 at a very high level. I want you to read these chapters, but I will not cover them in their entirety in class.

We will skip Chapter 4 (Processor Architecture), 5 (Performance Optimization), and 9 (Measuring Execution Time). Chapter 4 is worth reading if you're interested in how a simple processor with an Intel-like instruction set is implemented. Chapter 5 is all about understanding how to make programs run faster. Chapter 9 is all about how to measure how fast programs run.