Introduction to Computer Systems

Syllabus

Web Page

http://www.cs.northwestern.edu/~pdinda/ics-f01

Instructor

Peter A. Dinda
1890 Maple Avenue, Room 326
847-467-7859
pdinda@cs.northwestern.edu
Office hours: Mondays, 2-4 pm or by appointment

Teaching assistant

Dong Lu
1890 Maple Avenue, Room 238
847-491-3706
donglu@cs.northwestern.edu
Office hours: Fridays, 3-5 pm or by appointment

Location and Time

1890 Maple Avenue, CS Department classroom, WF 10:30-12am

Prerequisites

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

Textbook

Randal E. Bryant and David R. O'Hallaron*, Introduction to Computer Systems: A Programmer's Perspective (Beta Draft)* (Required - Textbook)

- This is a unique new textbook that will reach the shelves in 2002.
- Each of you will be receiving a free copy.

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

- 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 three 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 final purpose is to bring you up to speed in doing systems programming in a low-level language in the Unix environment.

It is important to note that this course is an import from Carnegie Mellon, and that **this particular iteration is a BETA TEST**. We will be using a beta draft of a fabulous new textbook and course materials. In addition to teaching you about systems, we want to evaluate these materials and this course in the context of our quarter system, both to provide feedback to the authors and to determine whether this should become a permanent course in our department. We promise that if you put in a reasonable effort, your grade will not be affected by the beta nature of this course.

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.

Resources

You'll be able to do the programming assignments on any modern Linux machine. We'll set you up with accounts on the TLAB machines. If you don't have an access card to room 125, you can get one from Pam Kearfott. It may also be possible to do the assignments using the Cygwin environment on Windows, but they will be graded in a Linux environment, so be sure your code works on the TLAB machines.

Labs

Programming labs will be periodically assigned. 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 can be done in groups of two.

Homework

Problem sets will be periodically assigned 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

Grading

10 %	Homeworks
50 %	Programming labs
20 %	Midterm (covers first half of the course)
20 %	Final (covers second half of the course)

Final grades will be determined by a combination of a curve and minimum thresholds. Peter ultimately assigns all grades. If you have a problem with a grade, you are welcome to bring it up with either Peter or Dong, but only Peter is empowered to change grades.

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
New student week				
1	9/26 W	Mechanics, Introduction, overview of abstractions using web request-response	Chapter 1	Data lab out
2	9/28 F	Physics, transistors, photolithography, Moore's Law, bits, bytes, and logic	2, 2.1, handout	
3	10/3 W	Integers and integer math	2.2-2.3	HW 1 out,
4	10/5 F	Floating point	2.4-2.5	
5	10/10 W	The Machine Model – instruction set architecture, microarchitecture, and basic instructions	3, 3.1-3.5, 5.7	HW 1 in, HW 2 out
6	10/12 F	Control flow	3.6	Data lab in Bomb lab out
7	10/17 W	Procedures	3.7	
8	10/19 F	Data	3.8-3.11	
9	10/24 W	Advanced machine code	3.12-3.16	
10	10/26 F	Memory and cache	6, 6.1-6.4	HW 2 in, HW 3 out
11	10/31 W	Cache performance	6.5-6.7	Bomb lab in,
12	11/2 F	Linking	Chapter 7	

Midterm exam, 11/6				
13	11/7 W	Exceptional control flow	8,8.1-8.4	Exploit lab out
14	11/9 F	Exceptional control flow	8.5-8.8	HW 3 in
15	11/14 W	Virtual memory	10, 10.1-10.6	
16	11/16 F	Memory system	10.7-10.8	Malloc lab out, *Buffer bomb lab in*
17	11/21 W	Memory allocation	10.9-10.12	HW 4 out
Thanksgiving break				
18	11/28 W	Input and Output	12, 12.1-12.4	
19	11/30 F	Network programming	12.5	
20	12/5 W	Concurrency	11, 11.1-11.4, handout	HW 4 in
21	12/7 F	Distributed systems + wrap-up	12.6-12.9, handout	Malloc lab in
Finals week				