COMP 264: Introduction to Computer Systems (Section 001)
Fall 2005 Course Information & Syllabus

Instructor:  R. I. Greenberg
Computer Science Department
Loyola University
Water Tower, Lewis Towers 512E
820 North Michigan Avenue
Chicago, Illinois  60611-2147

Phone:  (312)915-7981   Email:  rig@cs.luc.edu   Home page:  http://www.cs.luc.edu/~rig

Sometimes lecture notes or a summary will be available on the web. Other than that, if you have to miss a class, get notes from another student; mine are typically pieced together from more than one place with a lot of metacomments, which makes it hard for anybody but me to follow them. Also get copies of any missed handouts (available on the web site). The handouts are numbered sequentially, starting with handout 0. On handout 0, you need to fill in some information and return it to me promptly so you can be on the email list and get access to the web site for the course.

Office Hours:  In Lewis Towers 512E: 11:15–12:30 on Monday and Wednesday. These are the guaranteed times to find me except as announced in advance. You should also be able to find me at lots of other times; feel free to look for me or make appointments.

Course Objectives:  This course is designed to provide students with an understanding of the hierarchy of abstractions and implementations that comprise a modern computer system. The course is particularly geared towards topics of interest to a programmer, i.e., topics that affect the performance, correctness, or utility of user-level programs. Since this investigation is best carried out using the C programming language, the course will include some instruction in C for programmers familiar with Java.

Prerequisites:  COMP 170. (COMP 271 could also be helpful.)


Course Requirements:  There will be homework, three tests, and a final. The weightings within the semester grade will be: Homework: 25%, Tests 1–3: 15% each, and Final exam 30%.

Homework:  Only homework turned in by the due date is guaranteed to be graded. Any special circumstances that cause difficulty in meeting the deadlines should be brought to the attention of the instructor in advance. Homework must be handed in at the beginning of class, since solutions may be handed out in the same class on occasion. Homework turned in to my mailbox will generally not be graded, since I do not check the box continually and cannot generally verify that homework was turned in before solutions were distributed or discussed in class. If you cannot turn in homework in person, you should put it under the door of my office.

Exams:  The schedule tentatively calls for three 50 minute tests in week 6, week 9, and week 13. The final exam is scheduled for 8:30–10:30 am on Monday, December 12.

Collaboration:  No collaboration is permitted on exams.  *Collaboration* on homework is acceptable, but *copying* is not!  (Safeguard your files and printouts.) You may discuss solution techniques with other students, but you must write up your solutions independently. If you obtain a solution through research, e.g., in the library, credit your source and write up the solution in your own words.
Tentative Course Outline and Approximate Schedule:

Recommended readings from the text are shown on a weekly basis. (When selected sections or subsections are listed, it is assumed that you will include the introduction of the corresponding chapter or section.)

1. (8/29) Administrivia, computer representation of information, program translation, basic computer organization. Sections 1.1–6.

   Information representation. Sections 2.1.1–6.

   Integer representation. Section 2.2.
   Integer arithmetic. Section 2.3.

4. (9/19) Floating point representation. Sections 2.4.1–4.
   Floating point operations and rounding. Sections 2.4.4–5, 2.5.

   Data manipulation. Sections 3.3–5.
   Control. Sections 3.6.1–4.

6. (10/3) Control continued. Sections 3.6.5–6.
   Exam I on Chapters 1–2.
   Procedures. Section 3.7.

7. (10/12) Arrays. Section 3.8.
   Heterogeneous data structures, etc. Sections 3.9–11.

8. (10/17) GDB, buffer overflow, etc. Sections 3.12–13, 3.16.
   Optimizing program performance: Intro. Sections 5.1–3.

   Exam II on Chapter 3.
   Instruction scheduling and pipelining. Section 5.7.

10. (10/31) Still more on performance optimization. Sections 5.8–11.
    Program profiling. Sections 5.15.1–2.
    Storage technologies. Section 6.1.

    Cache memories Sections 6.4.1–4.
    Additional cache details and cache friendly code. Sections 6.4.5–7, 6.5, 6.7–8.


    Virtual memory continued. Sections 10.4–6.

15. (12/5) Additional material from Chapters 8 and/or 10 to be determined as time permits.