CS-2011 — Machine Organization and Assembly Language

Professor Hugh C. Lauer

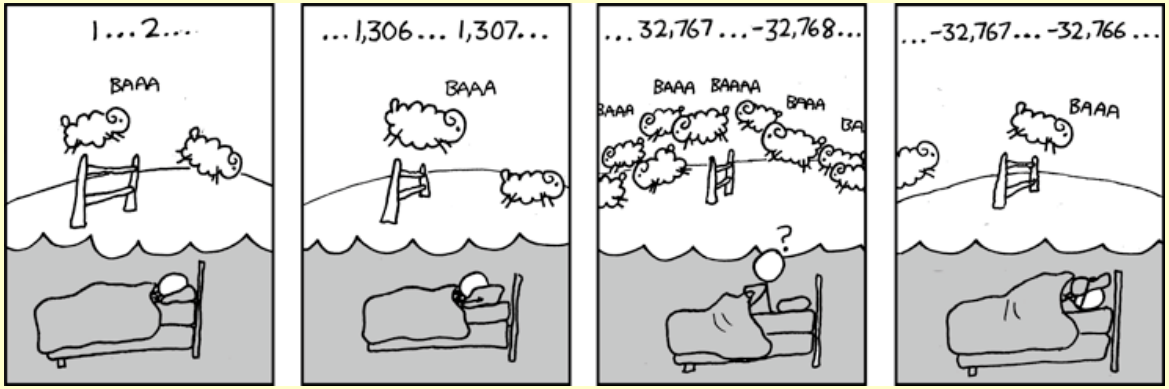
CS-2011, Machine Organization and Assembly Language

Overview

- A tantalizing problem (or two)
- Course introduction and logistics
- Course Survey and Photos
- Datalab assignment
- Course overview and theme

Two Tantalizing Problems

- **Is $x^2 \geq 0$?**
  - Float's: Yes!



  - Int's:
    - 40000 * 40000 → 1,600,000,000
    - 50000 * 50000 → ??

-352,516,352

- **Example 2: Is (x + y) + z = x + (y + z)?**
  - Unsigned & Signed Int's: Yes!
  - Float's:
    - (1e20 + -1e20) + 3.14 --> 3.14
    - 1e20 + (-1e20 + 3.14) --> ??

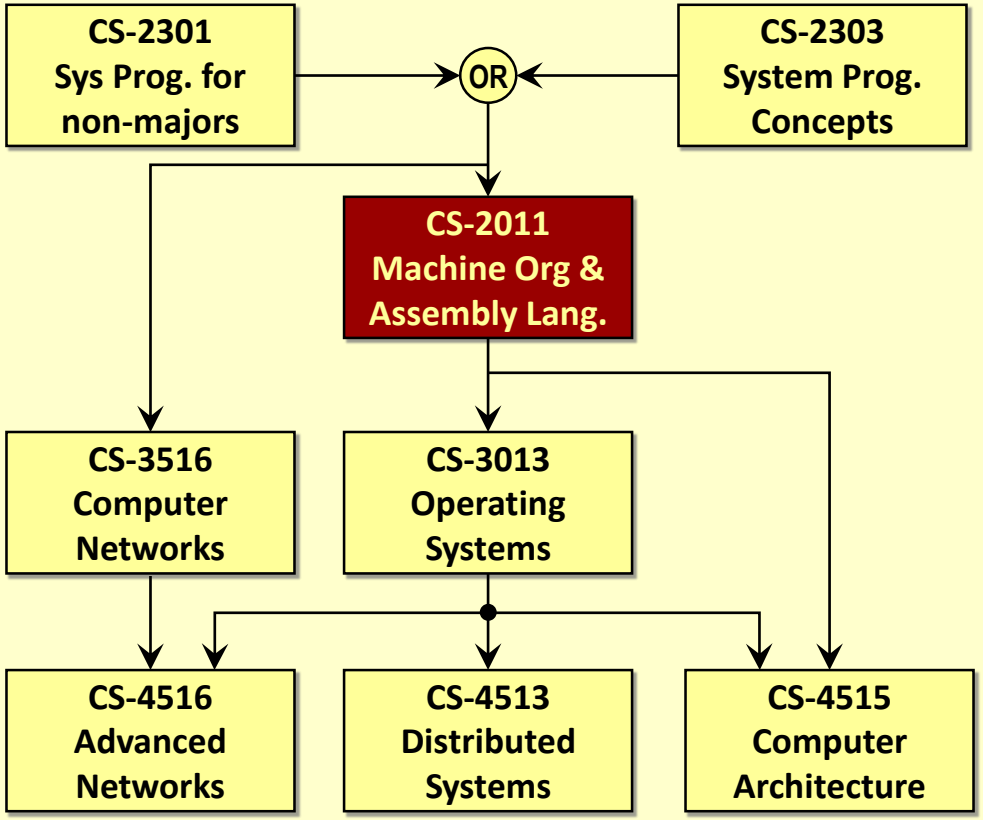
Computer Arithmetic

- **Does not generate random values**
  - Arithmetic operations have important mathematical properties
- **Cannot assume all "usual" mathematical properties**
  - Due to finiteness of representations
  - Integer operations satisfy "ring" properties
    - Commutativity, associativity, distributivity
  - Floating point operations satisfy "ordering" properties
    - Monotonicity, values of signs
- **Observation**
  - Need to understand which abstractions apply in which contexts
  - Important issues for compiler writers and serious application programmers

This course

- **Understand how numbers are represented in modern computers**
  - And the inner workings of arithmetic!
- **Read and work with Intel x86 and x86_64 assembly language**
- **Understand how C compiles to machine code**
- **Understand how modern computers work**
  - At 2000-level course

Role within CS Curriculum



Course perspective

- **Most Systems Courses are Builder-Centric**
  - Computer Architecture
    - Design pipelined processor in Verilog
  - Operating Systems
    - Implement large portions of operating system
  - Compilers
    - Write compiler for simple language
  - Networking
    - Implement and simulate network protocols

Course perspective (continued)

- **This course is programmer-centric**
  - Purpose is to show how by knowing more about the underlying system, one can be more effective as a programmer
  - Enable you to
    - Write programs that are more reliable and efficient
    - Incorporate features that require hooks into OS
      - E.g., concurrency, signal handlers
  - Not just a course for dedicated hackers
    - We bring out the hidden hacker in everyone
  - We cover material in this course that you won't see elsewhere

Course components

- **Lectures**
  - Higher level concepts, applied concepts, clarifications of texts, etc.
- **Labs (i.e., programming) projects — 4**
  - The heart of the course
  - 1-2 weeks each
  - Provide in-depth understanding of an aspect of systems
  - Programming and measurement
- **Recitation sessions**
  - Get started on Lab projects
  - Work through problems and details interactively
- **Weekly quizzes (5 small + 1 medium)**
  - Test your understanding of concepts & mathematical principles
  - Work out problems from textbook

Lab rationale

All projects/labs programmed in *C*

- **Each lab has a well-defined goal such as solving a puzzle or winning a contest**
- **Doing the lab should result in new skills and concepts**
- **We try to use competition in a fun and healthy way**
  - Set a reasonable threshold for full credit
  - Post intermediate results (anonymized) on Web page for glory!

More About This Course

- **This is *not* a traditional course in which**
  - We feed you a bunch of facts and methods
  - We assign you homework and quizzes to demonstrate that you know those facts and methods …
  - … and how to apply them!

- **Instead, we assign you problems that you *don't* know how to solve**
  - It is *your* responsibility to study them early, and …
  - … to figure out what it is that you don't know but need to know, and …
  - … to ask questions

More About This Course (continued)

- … Ask us
  … Ask the textbook
  … Ask each other

- Share your knowledge
- Raise questions and have discussions at the start of any class
- ….

More About This Course (continued)

- According to feedback in Course Evaluations:–
- This is the most difficult course that many students have encountered so far at WPI
- This course requires the most work and time of any encountered so far at WPI
- The course is the most fun of any course encountered so far at WPI
- At the end of the course, I hope tell you that …
- … you ain't seen nothin' yet!

Logistics

- **Lectures:–**
  - Monday, Tuesday, Thursday, Friday, 9:00-10:00 AM
  - Salisbury 115
  - No classes during Thanksgiving break, November 22-24,
  - Also, no class on Tuesday, December 12 ("Reading Day" or makeup day for snowstorms)
  - Quizzes:– Fridays November 3 – December 15
  - At *start* of class time. Approx 20-25 minutes (except last quiz)
    - You may begin as soon as you arrive in classroom
    - Stay in seat when you are finished

Anyone needing to take quiz at EPC should start early, be back in class at 9:20 AM

- **No make-up quizzes!**
  - Best four out of six
  - See Professor in advance if you must be away — *e.g., interviews, projects, etc.*
  - Let me know ASAP if you are sick or injured

Last quiz is mandatory in order to pass course!

Logistics (continued)

- **Recitation sections (called "labs" by Registrar)**
  - 9:00 AM, 10:00 AM (Zoo Lab)
  - 11:00 AM, 12:00 noon (Salisbury 123)
- **Attendance Counts!**
- **Must attend your own, registered session**
  - Space not guaranteed in later sections if you oversleep!
- **Waitlist:–**
  - Please bring an ADD-DROP slip *ASAP*!

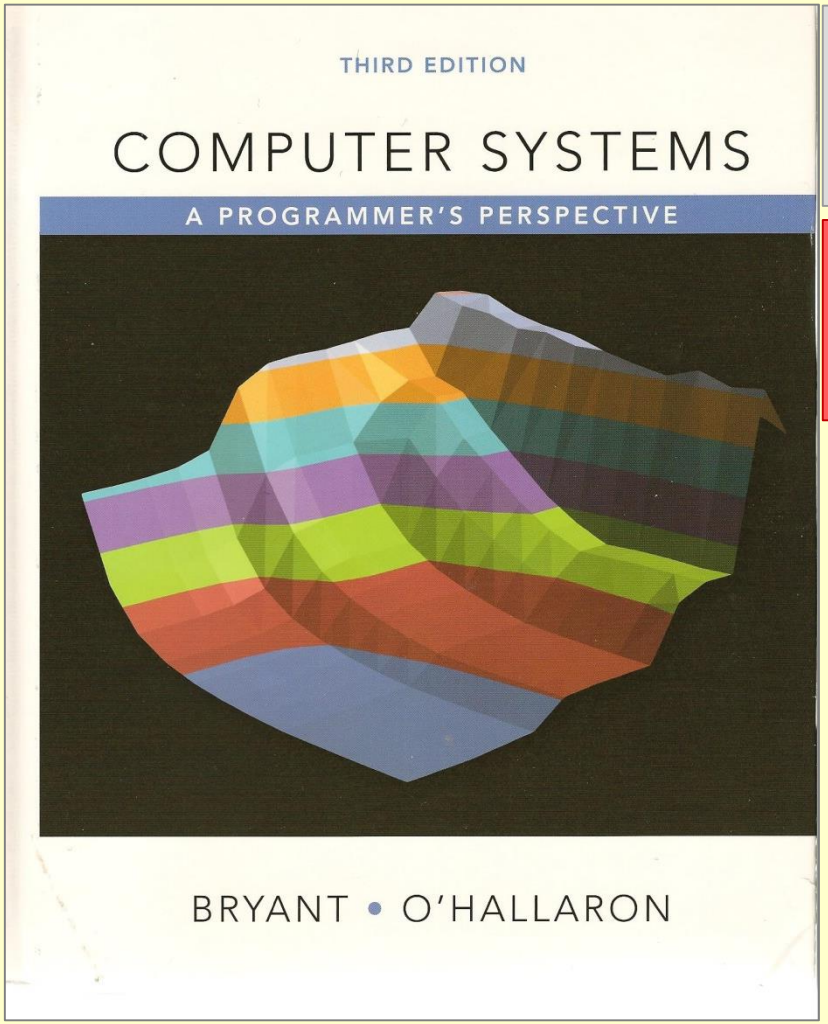
Questions?

Textbooks

- **Randal E. Bryant and David R. O'Hallaron,**
  - "Computer Systems: A Programmer's Perspective, Third Edition" (CS:APP3e), Prentice Hall, 2016
  - http://csapp.cs.cmu.edu
  - This book really matters for the course!
    - How to solve labs
    - Practice problems typical of quiz problems

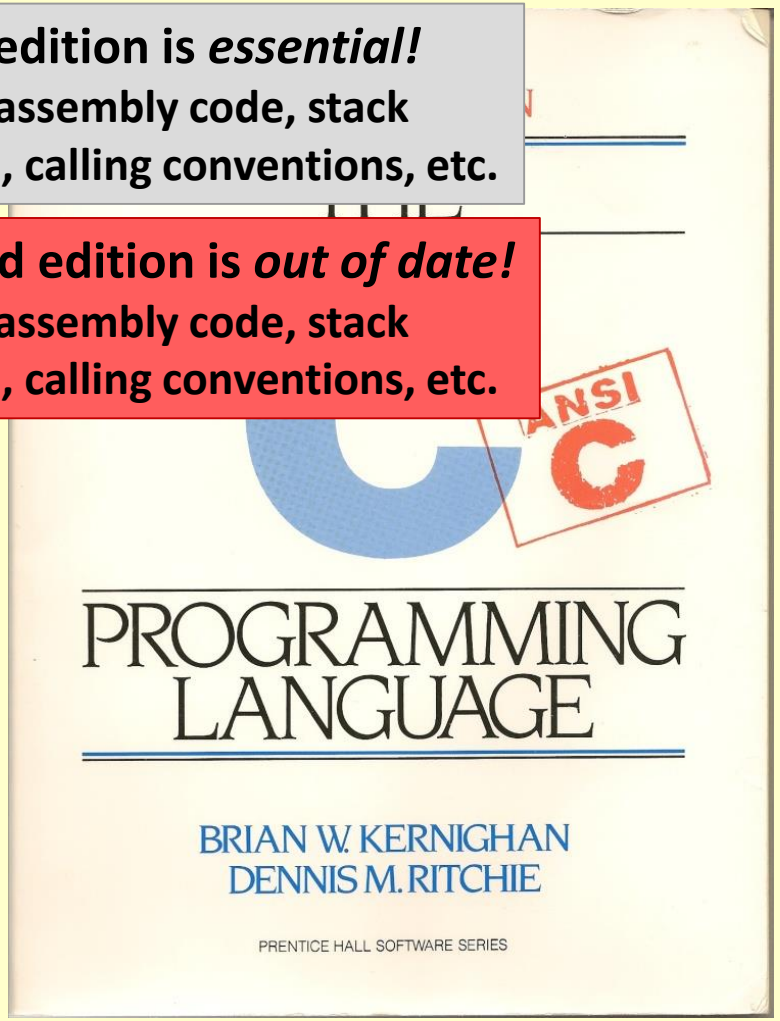
- **Brian Kernighan and Dennis Ritchie,**
  - "The C Programming Language, Second Edition," Prentice Hall, 1988
  - You should keep a copy of this on your desk for the rest of your (professional) life!

Textbooks (continued)

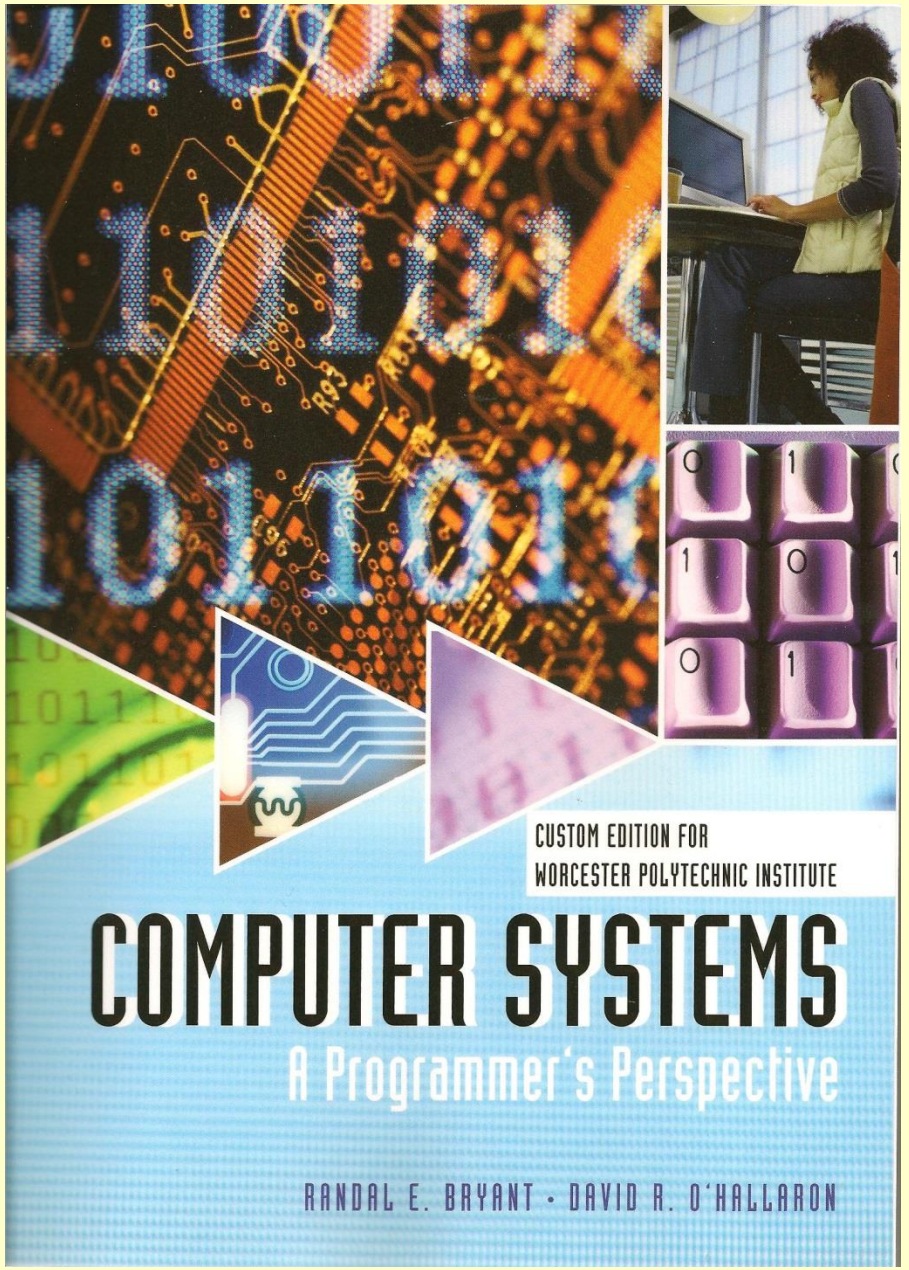


Third edition is *essential!*
64-bit assembly code, stack format, calling conventions, etc.

Second edition is *out of date!*
32-bit assembly code, stack format, calling conventions, etc.



Custom 3rd edition (chapters 1, 2, 3 and 5, 6, 7)



Tomorrow's Recitation Session

Introduction to Datalab

TAs and SAs

Rachel Plante

Teaching staff

Hugh C. Lauer
Adjunct Professor

- **Ph. D. Carnegie-Mellon, 1972-73**
  - Dissertation "Correctness in Operating Systems"
- **Faculty at University of Newcastle upon Tyne, UK**
- **30+ years in industry in USA**
- **WPI since 2006**
- **21 US patents issued**
- **2 seminal contributions to Computer Science**

Reading Assignment — Chapter 1 of Bryant and O'Hallaron

- We will assume full knowledge of this chapter throughout the course
- There may be questions on the contents of this chapter on *any* quiz

Grading

- **40-45% quizzes**
  - Final quiz is worth 2 times any other quiz
- **40-45% Lab projects**
  - Roughly equal in weight
- **10-20% Class and Recitation participation**
  - Helping each other
  - Contributing to discussion groups
  - Asking questions & asking for help
  - It is in your interest that the Professor and TAs know you by name
  - Please don't be embarrassed/annoyed if we ask you often

Getting Help

- **Class Web Page: http://www.cs.wpi.edu/~cs2011/b17**
  - Complete schedule of lectures, exams, and assignments
  - Copies of lectures, assignments, exams, solutions
  - Clarifications to assignments

- **Canvas**
  - Repository for copyright materials (lecture notes)
  - Linked directly from course web page
  - Submissions (backup for course servers)

- **Lecture Capture**
  - Lectures will be recorded using WPI's Echo Lecture Capturing system.
  - Link on Canvas home page for this course

Getting Help

- **Course mailing list: cs2011-all@cs.wpi.edu**
  - Use as discussion list
- **Staff mailing list: cs2011-staff@cs.wpi.edu**
  - Use this for ALL communication with the teaching staff
  - Send email to individual instructors or TAs only to schedule appointments
- **Office hours (Prof. Lauer):**
  - Monday, TBD
  - Tuesday, TBD
  - Thursday, TBD
  - Friday, TBD
- **1:1 Appointments**
  - You can schedule 1:1 appointments with Professor or any of the TAs

Ground Rule #1

- There are no “stupid” questions.
- It is a waste of your time and the class’s time to proceed when you don’t understand the basic terms.
- If you don’t understand it, someone else probably doesn’t it, either.

Ground Rule #2

- **Help each other!**
- **Even when a project or assignment is specified as *individual*, ask your friends or classmates about stuff you don't understand.**
- **It is a waste of your time try to figure out some obscure detail on your own when there are lots of resources around.**
- **When you have the answer, *write it in your own words* (or own coding style).**

Names and Faces

- **It is *in your own interest* that I know who you are.**
  - Class picture — will circulate next time
- **Students who speak up in class usually get more favorable grades than those who don't**
- **When speaking in class, please identify yourselves**

Policies: Assignments (Labs) And Exams

- **No work groups or project teams**
  - You must work alone on all assignments (unless otherwise specified)
- **Canvas**
  - Assignments due at 6:00 pm on the due date
  - Electronic turn-ins using *Canvas* (unless otherwise specified)
- **Conflict exams, other irreducible conflicts**
  - Make PRIOR arrangements with Prof. Lauer
  - Notifying us well ahead of time shows maturity and makes us like you more (and thus to work harder to help you out of *your* problem)
- **Appealing grades**
  - Within 7 days of completion of grading
  - Labs: Email to the staff mailing list
  - Quizzes: Talk to Prof. Lauer

Facilities

- **Lab projects to be completed on 64-bit Ubuntu virtual machines**

- **gcc for compilation**
  gdb for debugging
  Eclipse or Code::Blocks for GUI
  - DDD no longer supported!

Some students have used `nemiver` as a gui debugger

- **You may use any other platform**
  - On your own!
  - Projects graded on Ubuntu virtual machines

Timeliness

- **Lateness penalties**
  - **5% per hour (first two hours)**
  - **10% per hour after that**
- **Catastrophic events**
  - Major illness, death in family, …
  - Formulate a plan (with your professor *and* academic advisor) to get back on track
- **Advice**
  - Once you start running late, it's really hard to catch up

WPI Honesty Policy

- **What is cheating?**
  - Sharing code: by copying, pasting, retyping, looking at, or supplying a *file*
  - Coaching: helping your friend to write a program, *line by line*
  - Copying code from previous course or from elsewhere on WWW
    - Only allowed to use code we supply, or from CS:APP website
- **What is NOT cheating?**
  - Explaining how to use systems or tools
  - Helping others with abstract design issues
  - Working out a problem together on a whiteboard, etc.
    - Diagrams of data structures — *not* code
    - Discussing outlines of algorithms *in English, Chinese, etc!*
- **…**

WPI Honesty Policy (continued)

- …
- **It is a *violation* of the WPI Academic Honesty Policy to submit someone else's work as your own.**
- **It is *not a violation* of WPI's Academic Honesty Policy to ask for help!**
  - Classmates, TAs, friends, mentors, …
  - Explanations of things you don't understand

Unauthorized hacking

- **Any hacking into server will be treated as equivalent to *Academic Dishonesty***
  - Reported accordingly to Dean of Students Office.

- **If you suspect a vulnerability in *any system*, come and get permission to investigate *BEFORE* actually trying to probe.**

Note on Quizzes

- **Open book, open notes**
- **Many students have electronic textbooks**
  - Kindle
  - Tablet
  - Laptop
- **Okay to consult electronic textbook, electronic copies of notes, lecture slides, etc.**
- **May *NOT* connect to network, web, etc.**

Topics for this course

- Programs and Data
- The Memory Hierarchy
- Performance
- ~~Exceptions Control Flow~~
- ~~Virtual Memory~~
- ~~Networking and Concurrency~~

Questions?