Outline

- Introduction
- Why we need this course – 5 great “realities”
- How the course fits into the CS/ECE curriculum
- How to excel in this course
- Course components, policies, and logistics

Abstract System Model

Course Theme: Abstraction Is Good, But Don’t Forget Reality

- Most CS and CE courses emphasize abstraction
  - Abstract data types, and simple machine models
- These abstractions hide complexity, but have physical limits
  - Need to understand details of underlying implementations – to debug and to optimize performance
- Useful outcomes from taking CSci2021
  - Become more effective programmers
    - Able to find and eliminate bugs efficiently
    - Able to understand and optimize program performance
  - Prepare for later “systems” classes in CS & ECE
    - Compilers, Operating Systems, Networks, Computer Architecture, Embedded Systems, etc.

Programmer’s Perspective

Hardware Perspective

We use C language in this course

Processor (CPU)

Cache Memory

Main Memory

Input/Output Devices
Great Reality #1:
Ints are not Integers, Floats are not Reals

- Example 1: Is $x^2 \geq 0$?
  - Float's: Yes!
  - Int's: $40000 \times 40000 = 1600000000$

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

Source: xkcd.com/571

Great Reality #2:
You've Got to Know Assembly

- Chances are, you'll never write assembly programs
  - Compilers are much better & more effective than you are
- But: understanding assembly is key to machine-level execution model
  - Behavior of programs in presence of bugs
    - High level language models break down
- Tuning program performance
  - Understand optimizations done / not done by the compiler
  - Understanding sources of program inefficiency
- Implementing system software
  - Compiler has machine code as target
  - Operating systems must manage process state
- Creating / fighting malware
  - x86 assembly is the language of choice!

Great Reality #3:
Physical Implementation of Memory Matters

- Memory is not unbounded/unlimited
  - It has limited size, and must be allocated and managed
- Many applications are memory dominated
- Memory referencing bugs (e.g. pointers) especially pernicious
  - Effects are distant in both time and space
- Memory performance is not uniform
  - Cache and virtual memory effects can greatly affect program performance
- Adapting program to characteristics of memory system can lead to major speed improvements

Great Reality #4:
Optimizing Program Performance is Nontrivial

- Even exact operation count does not predict performance
  - Easily see 10:1 performance variation depending on how code written
  - Must optimize at multiple levels: algorithm, data representations, procedures, and loops
- Must understand system to optimize performance
  - How programs compiled and executed
  - How to measure program performance and identify bottlenecks
  - How to improve performance without destroying code modularity and generality

Great Reality #5:
Computers do more than execute programs

- They need to get data in and out
  - I/O system critical to program reliability and performance
- They communicate with each other over networks
  - Many system-level issues arise in presence of network
    - Concurrent operations by different processes
    - Coping with unreliable media
    - Cross platform compatibility
- Complex performance issues
Course Perspective

• This Course is Programmer-Centric
  • 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
    • You will know how to improve program performance, debug your programs more effectively

CSci2021 Role within CSE Curriculum

Transition from Abstract to Concrete!
• From: high-level language model
• To: underlying software and hardware implementation

Computer Architecture & Compiler

A few 8000-level courses

- Parallel Architecture
- Embedded Processor
- Virtual Machine
- Advanced Compiler Tech.
- CSCI 5004 Computer Architecture
- CSCI 5161 Compilers

How Do I Excel in This Course?

• This is a very large lower-division required course
  • We have ~120 students with very diverse backgrounds
• Lectures will cover only core course materials
  • Stay close to textbook
  • Cover most important topics and course materials
  • Attend all lectures to learn core ideas – save time
  • Read textbook before class to get deeper understanding
• The course has several components
  • 3 lectures per week
  • 1 recitation session per week (Thursday)
  • 4 homework assignments, 5 lab assignments, 2 midterms, and Final Exam

How Do I Excel in This Course? (cont.)

• Recitation Sessions (Thursday)
  • Clarification of lectures with more details
  • Elaborate key concepts with more examples;
  • Learn important tools and skills for programming assignments;
  • Explain details about lab assignments.
• Office Hours
  • Office hours every weekday (posted on class web page)
  • Ask specific questions
  • Help with labs and homework assignments.

Class Web Pages

Departmental Class Web Page:
• Public access without login
• Course syllabus, schedule, lecture notes, other useful info
  http://www-users.cselabs.umn.edu/classes/Spring-2016/csci2021/
• You should read course syllabus carefully for all course policies, personnel info, and course logistics.

Moodle 2.8 Web page:
• Use your university x500 account to access
• Class forums, assignment submission, exam info
  https://ay15.moodle.umn.edu/course/view.php?id=13068
Textbooks

The textbook:
  http://csapp.cs.cmu.edu
• Paper vs. electronic version
  Note: no e-version allowed during exams

Any Good “C” Book:
• Many free tutorial web sites available.
• Check class dept web site, under “Useful” info link.
• C language is closer to machine language than C++ and Java

Lecture Slides

• Lecture slides will be available on the course webpage before class, however:
  • They are incomplete, and thus
  • You must come to class to find out what is missing.

• The same rules also apply to recitation slides, except for
  • Recitation slides are available after recitations

  You cannot survive by just reading the lecture slides !!!

Grading Policy

• 4 Homework Assignments (10%)
• 5 Labs (30%)
• 2 Midterms (20%)
  • 2 50-minute in-class midterms; All exams are open book.
• Final Exam (40%)
  • Covering entire course, more weight on course materials after the 2nd midterm
  • Final Exam on 1:30pm-3:30pm, Wednesday, May 11, 2015
• Some guidelines:
  • ≥ 90% A−; ≥ 80% B−; ≥ 70% C−
  • Curve will likely to apply, in your favor, so the grade distribution will be similar to historical averages.

Policies: Assignments, Labs, and Exams

• Group? No
  • You must work alone on all assignments
• Submission process
  • Labs due online, by 11:55pm on due date, submission online
  • Homeworks due on paper, preferred printed, submission before the lecture on due date
  • Late submission for 1 day loses 15%, no credit after one day.
• Schedule conflicts
  • No make-up exams
  • One excused exam will be replaced by more weight on final
• Appealing grades
  • Within 7 days after the completion of grading (see Syllabus for details)

Cheating

Cheating will NOT be tolerated!!!

• What is cheating?
  • Sharing code: either by copying, retyping, looking at, or supplying a copy of a file.
• What is NOT cheating?
  • Helping others use systems or tools.
  • Helping others with high-level design issues.
  • Helping others debug their code.

Class Forums and UNITE Streaming Video

• Class forums
  • There is a discussion forum on Moodle web page
    • You are welcome to discuss programming assignments and homework assignments there
  • Course staffs will monitor the activity of this forum closely, and will answer questions that are post there in a timely manner
• UNITE streaming video
  • Available after 10 days
  • Link will be available on class web page.
Lab 0 – Logistic Practice

- Learn how to log into Unix machine, edit and compile a program
- “Hello, world”-style program that just prints a message
- No need to turn in Lab 0
- More details and other key C programming language features are covered in tomorrow’s recitation sessions.

Next Lecture ...

- Data Representation I:
  - Bits and Bytes
  - Binary numbers
  - Integer representation

WELCOME AND ENJOY!