



Introduction

ICS312 Machine-Level and Systems Programming

Henri Casanova (henric@hawaii.edu)



Course Goal

- Overall goal: Understand many of the things that happen “under the cover” when running a program on a computer
 - There is a lot of complexity there, as you can imagine
- By the end of the class you will
 - Have a basic understanding of how a CPU works and good understanding of how it can be programmed at a low level
 - Be able to write programs (or pieces of programs) in x86 assembly
 - Understand the concepts behind compiling, linking, loading, and debugging
 - Have implemented a tiny compiler

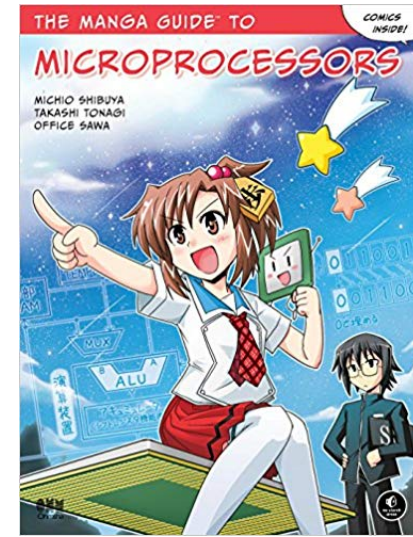


Course Website

- Located at:
 - github.henricasanova.io/ics312_spring2021
 - Linked from my personal homepage
 - Google for “Henri Casanova”
- Organized as Modules
 - All lecture notes as PDF files
 - Pointers to useful on-line material
 - All assignments
 - Announcements
 - A link to the Syllabus
 - Which we’re going over now in these slides
- Let’s look at the Web site...

Textbook(s)

- The main text is a **free** book:
 - *PC Assembly Language*, Paul A. Carter
 - Available for download on the course's Web site
- The user's manual for our assembler, NASM, is also available on the course's Web site, and other manuals will be made available throughout the semester
- Other interesting free resource available for download:
 - *The art of assembly programming*, John W. Lockwood
 - *Assembly Language for x86 Processors*, Kip Irvine
- See links from the Syllabus on the Course Web site
- A former ICS 312 student highly recommended this textbook: The Manga Guide to Microprocessors
 - I am not affiliated with the author/publisher in any way and I have not read it





Lectures and Office Hours

- Lectures will be via Zoom
- Lecture notes are posted on the course's Web site regularly
 - You can read them before or after the lecture, up to you really
 - I am notorious for spacing out on putting the notes up on the site, so just drop me a one-line e-mail
- Practice Quizzes are also posted regularly on Laulima
- Office hours, teaching assistant, etc
 - See information on the Web site

Inverted Lectures

- A *few* lectures will be “inverted”
 - You watch a screencast at your own pace
 - The lecture period is for questions and practice exercises
- I do this for a few topics in the course that are more “mechanical” or “difficult”
- You must watch the screencast ahead of time!
 - E-mails reminders will be sent out
- Scheduling may be imperfect
 - Out-of-order and/or overlapping modules
 - We might end a few lecture periods early



Screencast Lectures

- A few lectures might be **screencast**
 - This is because I am often required to travel to conferences or to the National Science Foundation this semester
- This will be announced with as much advance notice as possible
 - And may not happen at all this semester due to the pandemic



Grading on 1000 points

- Two exams
 - One midterm exam (250 points)
 - One cumulative final exam (250 points)
- Nine homework assignments
 - Homework #1: 16 points
 - Homework #2: 56 points
 - Homework #3: 48 points
 - Homework #4: 40 points
 - Homework #5: 90 points
 - Homework #6: 80 points
 - Homework #7: 60 points
 - Homework #8: 70 points
 - Homework #9: 40 points

Exams

- Exams will be taken on Laulima
- There is a single “question” that has the whole exam as a PDF file
 - You can view/download/print
- You provide your answer as a single text-file that you upload into Laulima
 - Plain text or PDF
- Exams are all open notes, open computer, open everything
- Of course they have to be done **individually**
 - They are fully randomized

Homework Assignments

- All assignments must be turned in electronically using Laulima by 11:55PM on the day the assignment is due
- **Late Assignments**
 - 10% penalty for up to 24 hours of lateness
 - A grade of zero for more than 24 hours of lateness
 - e.g., if the due date is 3/10, an assignment turned in at 1AM on 3/11 will be penalized by 10%, and given a zero if turned in at 5PM on 3/12
- **Solutions will always be discussed in class, and available on Laulima**
- Read the syllabus' statement about "academic dishonesty"

Homework Assignments (cont.)

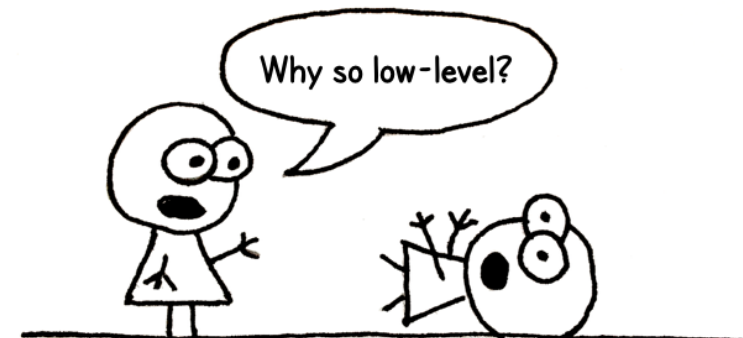
- All assignments are individual (no group assignments in this course)
 - Some assignments will be “pencil-and-paper” and require no programming
 - Some assignments will be programming assignments
 - Both types of assignments can overlap in time
- Rough point breakdown for the 9 assignments:
 - 30, 40, 40, 30, 90, 80, 70, 80, 40

Practice Quizzes

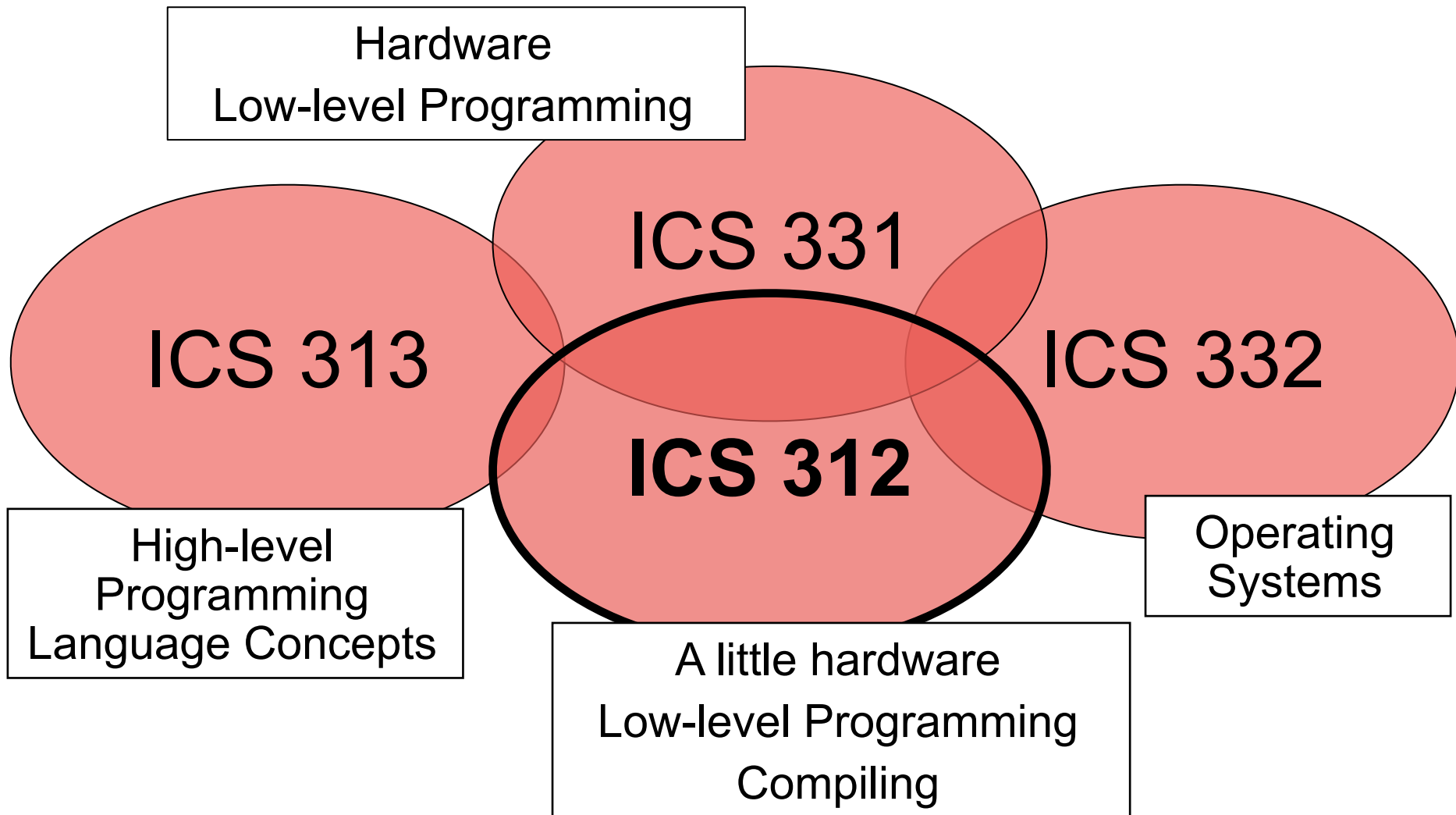
- When I teach this course f2f, we have about 8 quizzes in the semester
- This is really hard to do on-line
- But experience shows that when I don't do the quizzes (in the f2f case), overall student grades on exams are much worse
 - Yes, you don't like quizzes, but for many of you they provide some motivation to keep up
- What we'll do:
 - I'll post practice, ungraded, quizzes
 - For most of them I'll go through answers in class via Zoom polls
 - It's up to you if you want to look at quizzes ahead of time, or just when I go through them in class
- The idea is that if you find that you'd answer wrong on a lot of the quiz questions, then you should catch up

What is this course about?

- At this point in your computer science education, most of you have only a very high-level understanding of how a computers run programs
- Your experience so far: you write code in some language (say a compiled language like Java, C/C++), you build it, and then you run it
 - Each of these steps involve hitting buttons in IDEs or typing some Shell commands
- You all know how to do this, but (most of) you have only very superficial understanding beyond “this magically creates an executable”, “this magically runs an executable that my CPU magically knows how to do”
- A big part of being an effective (and employable) computer scientist is knowing what the magic is



ICS312 and the ICS Curriculum





Topics in this class

- What is inside a computer?
 - Elements of computer organization
 - Elements of computer architecture
- Assembly programming
 - Intel x86 Assembly
- What is compiling, linking, loading, and debugging?
- How does one build a compiler?
 - We'll build a small one
- **This is a hands-on course**
 - Writing (pieces of) programs in x86 assembly
 - Tiny bits of C (to be read) and Java (to be written)

How to not do well in this course?

- **Don't come to class (“the slides are nice”)**
 - We do a LOT of stuff in class, including live coding, and I give a lot of explanations, examples
- **Start assignments late (“I work better under pressure”)**
 - Assembly programming and compilers are difficult topic
 - Starting late seems to be a growing trend, and it's a problem
 - Read the assignment early to subconsciously start thinking about it
 - **We will not answer assignment-related e-mail on the day it's due**
- **Don't come to office hours (“The instructor is scary because he shows ‘how to no do well in this course?’ slides”)**
 - After you struggle for a while on something, drop by
 - But don't expect to “camp” in the office hours for the solutions to be given out
 - Instructor and TA office hours are an amazing service provided to you, and yet, they go mostly unused



How to not do well in this course?

■ Cheat

- Last time I taught this course, 2 students got an F due to cheating, and 3 got a C instead of an A due to cheating
- Cheating seems to be on the rise, and consequences will become more and more dire department-wide
- This is also part of the reason for having 50% of the course's points being exams

■ Aim for a “passing grade”

- This is a course in which, if you apply yourself, you should easily get a B or above
- Yet, I typically give way too many C's
- Something to think about...



Show of hands

- To get an idea of your backgrounds I've got a few “show-of-hands” questions
 - It's totally OK if all the answers are “No”, don't panic
- Who has taken / is taking ICS331?
- Who has taken / is taking ICS313?
- Who has written assembly code before?
- Who is using the Shell (Linux / macOS) regularly?
 - We don't need much in this course
 - Note that this is something that you will have to do way more than you think beyond graduation
 - The “Getting Started” module has some pointers (let's look at them quickly)

Software/Hardware for ICS312

- You'll have to use an Intel-based machine that speaks the x86 32-bit Instruction Set Architecture (IA-32)
 - Luckily, we all have that
 - In fact we have 64-bit machines!
 - But in this course I use 32-bit assembly because:
 - It has less “stuff” and is better for a first experience
 - We learn exactly the same set of things
 - Our free textbook is about 32-bit assembly
 - When you'll need to go to 64-bit assembly, the transition will be easy
- We'll use the NASM software package, which is also free
- Let's look at **Homework Assignment #0**, which is ungraded but which you should do as soon as possible in the semester
 - Don't show up the day the first programming assignment is due with a “I can't run NASM” problem!



More Questions

- Any questions on the syllabus?
- Any questions on the course in general?

What's Next?

- There is already an announcement on the course Web site regarding **two short screencasts to watch before this coming WEDNESDAY**
- These screencasts are about “numbers and computers”
 - Many of you will find them easy and can fast forward a lot of the material
 - Make sure you watch them before the Monday lecture next week as we'll do in-class practice exercises to make sure we're all up to speed
- This is all in the Integers and Computers module... let's look at it now...
- And don't forget to get started on **Homework #0...**