Introduction

ICS312 - Spring 2014
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
  - Turns out there is a lot of complexity there
- By the end of the class you will
  - Have a basic understanding of how a CPU works and can be programmed
  - Be able to write programs (or pieces of programs) in x86 assembly
  - Have a basic understanding of compiling, linking, loading, and debugging

Course Website

- Located at:
  - http://navet.ics.hawaii.edu/~casanova/courses/ics312_spring14
  - Linked from my personal homepage
    - Google for "Henri Casanova"
- Contact: henric@hawaii.edu
- Contains
  - All lecture notes as PDF files
  - Pointers to useful on-line material
  - All assignments
  - Announcements
  - A link to the PDF Syllabus
    - Which we're going over now in these slides

Textbook

- 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
- Another interesting free resource available for download is:
  - *The art of assembly programming*, John W. Lockwood
- Do not print the above on ICS printers, print them at your own expenses (e.g., using ITS)
- Finally, an interesting book that could come in handy for more details is:
  - Assembly Language for Intel-based Computers (5th edition), Kip Irvine
  - Available from Amazon for ~$100 (new) or ~$50 (used)

Lectures and Office Hours

- Lecture notes should be 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 my an e-mail
- Office hours
  - Wed, 2PM-4PM
- Teaching Assistant: Robert Namahoe
  - POST 303-1
  - Mon/Wed, 11AM-2PM
  - e-mail: rnamahoe@hawaii.edu

Inverted Lectures

- *Some* lectures will be "inverted"
  - You watch the video of the lecture on your own time, so that you can easily pause to absorb the concepts
  - The lecture period is used to ask questions and do practice exercises
- I do this for those topics in the courses that are more "mechanical" than "conceptual"
  - Some might say "dry"
- It is paramount that you watch the video ahead of time, otherwise there will be no benefit and you'll bomb on the tests!
- E-mails reminders will be sent out of course
- Since scheduling may be imperfect (i.e., not all sets of lecture slides end exactly at the end of one lecture period), there could be a little bit of harmless but "out of order" content delivery

Homework Assignments

- Assignments
  - All assignments must be turned in electronically (e-mail to the instructor and the TA) by 11:55PM on the day the assignment is due
- Late Assignments
  - 10% penalty for 1 day of lateness
  - 50% penalty for 2 days of lateness
  - A grade of *zero* for more than 2 days 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 by 50% if turned in at 5PM on 3/12
- Solutions will always be discussed in class, and available upon request by e-mail to me
- Read the syllabus' statement about "academic dishonesty"

Exams, Grading

- Exams
  - In-class quizzes (about 6 in the semester)
    - Always on "this coming Tuesday"
  - Midterm #1
  - Midterm #2
  - Comprehensive final exam
- Grading

Homework Assignments	50%
Quizzes	10%
Midterm #1	10%
Midterm #	10%
Final	20%

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 computer runs programs:
  - You write code in some language (Java) and compile it
    - Compilation is probably a "magical" step at this point, and you're using an IDE that completely hides it
  - You run the code
    - Somehow the computer "magically" runs stuff
  - You debug with print statements and the like
  - You repeat
- A lot of magic happens here under the cover
- A big part of being a good computer scientist is knowing what the magic is, which is what we'll learn in this class

Topics in this class

- What is inside a computer?
  - Elements of computer organization
  - Elements of computer architecture
- Assembly programming
  - Intel Assembly
- What are compilers, linkers, loaders, and debuggers and how do they work?
  - Main principles, some examples
- A hands-on course
  - Writing (pieces of) programs in assembly code
    - And a tiny little bit of C

ICS312, ICS331

- ICS312: How does one write programs at the machine level
- ICS331: How does one design components inside a microprocessor
- ICS431: Computer architecture
- All three courses have to do with Computer Architecture
  - What does the inside of a computer look like?
  - What's a good way to build a computer?
- So we'll start the semester with a simple review of Computer Architecture principles
  - Something we also do in ICS332 (Operating Systems)

Show of hands

- To get an idea of your backgrounds I've got a few "show-of-hands" questions, bear with me
  - It's totally OK if all the answers are "No", don't panic

- Who has taken / is taking ICS331?
- Who has taken / is taking ICS332?
- Who has taken / is taking ICS431?
- Who has written code in Assembly before?
- Who has written code in C before?
- Who has used Linux/UNIX before, even if only a little bit?

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 pretty much all have that...
- We'll use the NASM software package, which is free
  - Everything's free in this course :)
- Let's look at Homework Assignment #0, which is ungraded but which you should do as soon as possible in the semester

That's it for today!

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