



Introduction

ICS432 Concurrent and High-Performance Programming

Henri Casanova (henric@hawaii.edu)

Course Goal

- A “hands-on” course on concurrency and high performance programming
 - There is a lot of theory that we could go into
 - Instead we’ll take a pragmatic approach and experience general principles
- By the end of the class you will be able to:
 - Write correct and efficient multi-threaded code in multiple languages (Java, C++, Python)
 - Understand how to accelerate code, in general but mostly via multi-threading
 - Have notions of advanced topics on concurrency and high performance if time permits (locality, lock-free programming, transactional memories, ...)

Course Website

- Located at:
 - http://courses.ics.hawaii.edu/ics432_fall2024/
 - Linked from my homepage
 - Google for “Henri Casanova”
- Organized as Modules
 - All lecture notes as PDF files, including some screencasts
 - **No textbook in this course!**
 - Pointers to useful on-line material
 - All assignments
 - Announcements
 - A link to the Syllabus
 - Which we’ll go over now in these slides
- Let’s look at the Web site...

Lectures

- Lecture notes are posted on the course's Web site regularly
 - You can read them before or after the lecture, up to you
 - I am notorious for spacing out on putting lecture notes up on the site, so just drop me a one-line message
- In general, if you note “anything weird” on the Web site or in the lecture notes, do not hesitate to write me a one-liner
 - Broken links, typos, weird “due date”, etc.
- Some modules in the course are a bit independent from the rest of the course, and we may cover them out-of-order
 - e.g., so that assignments are more spaced out

What is this course about?

- Focus #1: **Concurrency**: write programs that do multiple things at once
 - Very mainstream and important
 - Known to be pretty difficult...
- Focus #2: **High Performance**: write programs that go fast
 - Always a good idea to know something about this
 - Concurrency is one of the ways to achieve high performance, but not the only one
- Focus #3: **Software Development**
 - This course is resolutely hands-on, and you will write a fair amount of code in a private GitHub repo that you'll contribute to throughout the whole semester, with significant software (re)engineering

Concurrency is Hard

- Writing concurrent programs is difficult when one has never been exposed to concurrency
 - And difficult even when exposed to concurrency!
- After taking this course you will be “”””proficient”””” in concurrent programming
 - Correctness, Responsiveness, Performance
 - Java, C/C++, a bit of Python
- Why so many quotes around proficient?

Herb Sutter, chair of the ISO C++ standards committee:
“Everybody who learns concurrency thinks they understand it .. and discovers that they didn’t actually understand it yet after all.”

Relationship to ICS Curriculum

- ICS332: Operating Systems
 - Provides an introduction to concurrency concepts
 - We'll have a review of these concepts (as a screencast)
 - This course expands **massively** on the “Synchronization” module of ICS332
- ICS443: Parallel Algorithms
 - About the theoretical aspects that are relevant in this course, but that we not got into since this is a “hands-on course”
 - And there is a distinction between “concurrent” and “parallel”, which we'll come to...

Shows of hands

- Who has written multi-threaded code not in a college course?
 - If so, which language/framework?
- Who has ever tried to make a piece of code faster (not counting ICS332)?
- Anybody would have difficulties bringing a laptop to class for some of the lectures?
 - I like to have in-class coding sessions where I go from student to student to provide help for challenging homework assignments

Homework Assignments

- The class will have **both** programming assignment and non-programming assignments
- All programming assignments are part of a **semester-long software development project**
 - An image-processing application to which we add features throughout the semester
- Non-programming assignments cover the same concepts as the programming assignments
 - Because “I have written code that happens to work each time I run it” does not equate “I have understood everything correctly” for concurrent programming

Assignment Specifications

- Each assignment has clear specifications with examples
 - File names, class names, behaviors, etc.
- Not conforming with specification makes our life (i.e., grading) very difficult
 - And it will make you absolutely hated by your co-workers in your professional lives
- **You will lose points for not matching the specs**
 - But if the specs are unclear, do complain!!!

Command-Line

- Use of the command-line in the terminal is a vital skill
- We are not doing anything fancy at all in this class on the command-line, but if you're not comfortable with it it's high-time you do something about it
 - A lot of graduates contact us back saying "I wish I had learned more command-line stuff while in college because my first month on the job was rough! I thought it was just a professor thing..."
 - <coffeeshop anecdote/>
- There is a Reading in this module about the command-line....

Software for ICS432

- Operating System that runs Java (whatever one)
- We need Java 21 and Maven 3.9.x
- Other software will be provided as part of Docker containers
 - And you will yourselves create and use a few Docker containers
 - Because that's another vital skill today!
- We'll look at Assignment #0 at the end of this lecture...

All Assignments are Individual

- All assignments in this course are individual
- The programming assignments in previous versions of ICS432 used to be team assignments
- But now we have ICS496 (Capstone) which is all about teamwork
- And in ICS432 no matter what I tried (fixed teams, randomized teams, shuffled teams), a fraction of the students never did enough on the assignments, and then bombed on the exams

Homework Assignments Policies

- All assignments are to be turned in 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
- **Solutions will always be discussed in class, and available upon request by e-mail to me**
- Read the syllabus' statement about "academic dishonesty"

Grading on 1000 points

- Two Exams
 - One midterm exam (250 points)
 - One cumulative final exam (250 points)
- Homework Assignments (500 points):
 - Homework #2: 40 points
 - Homework #3: 55 points
 - Homework #4: 30 points
 - Homework #5: 50 points
 - Homework #6: 50 points
 - Homework #7: 40 points
 - Homework #8: 55 points
 - Homework #9: 50 points
 - Homework #10: 50 points
 - Homework #11: 45 points
 - Homework #12: 35 points

Professor Advice #1

- Come to class
- I've been told my slides are nice and clear, and so no need coming to lectures
- That is true for about 5% of students, and a horrible mistake for the rest
- Concurrency is known for many “I get it”, “wait, not I don't”, “Oh, I got it!”, “What??? No, I didn't!”
- Flipping through slides without the in-class explanations/digressions/live-coding lures most students into a false sense of “getting it”

Professor Advice #2

- Ask questions, including “can you explain this again?”
- 95% of students at UH think: “I must be the only student who’s confused and it’s so embarrassing to speak up”
- But concurrent is confusing to most people! (see previous slide)
- I really want to avoid is what has happened so often:
 - 5% “vocal” students are getting the material so well they only ask questions that go beyond the course material
 - 95% “mute” students are not getting the material, and never ask a question (and are understandably intimidated by the “seemingly crazy” questions from the vocal students)
- Not sure what do do about this, but I want the “crazy” **and** the “I didn’t understand any of this” questions

Professor Advice #3

- Don't start late on assignments
- Every professor tells you this, and it's typically ignored
 - Because most of you have a lot on your plate, we get it
- But at the 400-level, courses become challenging and assignments can be tricky/long, so starting late becomes more and more dangerous
 - Some of you can still blast through assignments in a day, but the fraction who can do so decreases dramatically at the 400-level
- **One good approach:** Read through the assignment on the day it is posted
 - So that you subconsciously think about it right away
 - I found this extremely useful as a student
- We will not answer any questions (including via e-mail) on the day the assignment is due



Questions?

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



Homework #0

- Homework #0 is about making sure you have all the software you need on your machine
- Let's look at it now....

Homework #1

- Homework #1 is about making sure you can run starter code for the semester-long project
- It's a stand-alone Java app with a GUI
- In this sense it's a bit "passé" (even though we run GUI apps on our machines all the time, and people are developing them every day!)
- BUT:
 - It's the perfect vehicle to teaching all our material in this course
 - I've implemented 95% of the GUI stuff anyway, so "GUIs in Java" is NOT a learning objective of this course
 - I've done a Web app version just to see: it's more complicated, weirdly convoluted, not a good way to get one's feet wet with concurrency
 - In the end, it's an annoying concurrent frontend and a nice concurrent Java backend, so a lot of hassle for essentially no pedagogic benefits!
 - So a Java GUI it is in this course
 - At least until Browsers/JavaScript/NodeJS/... become good-for-teaching-concurrency
- Let's look at it now....



What Now?

- Do Homework #0 and Homework #1 ASAP
 - At the latest by the end of next week
- Come with questions/issues right away so that we can get you all set up...

What's Next

- Do Homework #0 and Homework #1 by the end of next week
- You may also want to:
 - Brush up on your Java soon, C/C++ later
 - Brush up on git if needed (hopefully not, right?)
 - [ICS314 Review Site](#)
 - Learn the basics of Maven if needed
 - <https://maven.apache.org/guides/getting-started/maven-in-five-minutes.html>
 - Instructions/examples will be provided in our assignments though