CS 241 Honors
Class Introduction

CS 296-41 Course Staff

University of Illinois Urbana-Champaign

January 23, 2018
The Goals

- To learn about systems programming at a deeper level than in CS 241 through lectures about various systems topics throughout the semester
- To explore systems programming by working on a semester-long project of your choice
About This Course

Lectures

- Tuesday, 7-7:50pm, 3403 Siebel
- Full lecture schedule will be available on the honors course website and on Piazza soon
- There is an attendance grade, but we will drop 2 lecture attendance without question
- If there is an extenuating circumstance, please contact us on Piazza.
Projects

- Teams consist of 2-3 people (see us for anything else)
- Each team will be assigned a mentor from course staff
- Every team will meet with their mentor once a week
- Must be technically rigorous
- Must relate to systems programming
- Games are highly discouraged
About This Course

Grading

- 15% – Lecture attendance
- 85% – Semester project
  - 5% – Project Proposal
  - 25% – Weekly check-ins with mentor
  - 35% – Final deliverables (code, project website, etc.)
  - 20% – Final presentation

The minimum grade thresholds for this class are the same as CS 241
<table>
<thead>
<tr>
<th>Name</th>
<th>NetID</th>
<th>Interests</th>
</tr>
</thead>
<tbody>
<tr>
<td>Aneesh Durg</td>
<td>durg2</td>
<td>Functional Programming, TextUIs</td>
</tr>
<tr>
<td>Kevin Hong</td>
<td>khong18</td>
<td>Concurrency, IPC, Statistics, Machine Learning</td>
</tr>
<tr>
<td>Jonathan Wexler</td>
<td>jwexler2</td>
<td>Parallelism, Networking</td>
</tr>
<tr>
<td>Shreyas Patil</td>
<td>srpatil2</td>
<td>Systems, Security, Android</td>
</tr>
<tr>
<td>Steven Shang</td>
<td>shang9</td>
<td>Distributed Systems, Cloud Computing</td>
</tr>
</tbody>
</table>

If there is someone you are interested in working with, let us know!
What is systems programming?
Project Ideas

The big topics in systems programming include...

- Process Control
- Memory Management
- Parallelism
- Networking
- Security
- Filesystems
Project Ideas

Process Control
- Managing the execution of programs on one (or multiple) machines
- Past Projects: Raspberry Pi Compute Cluster, Distributed Server Manager, earliest-deadline-first process scheduler

Memory Management
- Managing/Abstracting process use of memory and resources
- Past Projects: Custom C Garbage Collector

Parallelism
- Using multiple processors to work together on a problem
- Past Projects: Chess Engine, GPU Based Fluid Simulator
Project Ideas

Networking
- Having remote computers talk to each other
- Past Projects: Remote Desktop, multi-user real-time text editor

Security
- Keeping undesired entities from accessing protected resources
- Past Projects: Custom Network Encryption tool, Build a Proxy Server, Steganography Toolset

Filesystems
- Organizing bits and bytes into readable data
- Past Projects: Distributed FUZE Filesystem
Brainstorming Time!
A couple other things about your projects...
Writing Readable Code

- When you write solutions for an MP, the only other entity looking at your code is a machine.
- Now, you are writing code that *other people* (including those grading your project) have to read!
- Low-Level languages like C can get especially hard to comprehend when written badly.
- Some general guidelines:
  - Name your variables and functions so that it’s obvious what they’re used for.
  - If someone else can’t tell what your code is doing at a glance, write comments to explain it.
  - Code in whatever style you want, but make sure your style is consistent.
- Keep in mind that your teammates are not mind-readers; they cannot guess what you were thinking when you were writing your code.
If you are working on a team, you will likely need to meet as a team (outside of your mentoring meetings) to determine who is doing what. Consider managing units of work on your project using the issue tracker on GitHub.

- Each logical unit of work should have its own ticket
- Using an issue tracker allows you to keep track of any conversation pertaining to a particular feature
- Assigning team members individual tickets can help organize collaboration

If two members are developing features that interact with one another, make sure you are coordinating your work so that your features can work together properly.
These projects are long-term and fundamentally different from most other assignments you’ve done.

It’s very easy to fall behind, so make sure you’re making progress every week.

Suggestion: make deadlines for yourself over the course of the semester! (or ask your mentor for deadline suggestions)

It’s much easier to make a few hours of progress every week instead of trying to make 40+ hours of progress in a single week.
Brainstorming

Hopefully you all have some initial ideas for projects you would like to work on! Today, you’ll split into groups and do the following:

- **Share project ideas**
  - What do you want to do?
  - Why?
  - How might you do it?

- **Give feedback on other’s ideas**
  - What about their idea is interesting?
  - What seems hard or challenging?
  - What could be improved or changed?

- **Synthesize project ideas**
  - Could you take ideas from another student’s project and apply them to your own?

- **Find potential teammates**
Submission Instructions
Submit your project proposal as a *private* post on Piazza with title “Project Proposal (netid1, netid2, netid3)” where netid1, etc., are the NetIDs of your group members.

Be sure to include your names in the body of the proposal.

Submission Deadline
Proposal submissions are due **Thursday, February 2, at 11:59pm CST**.

Feel free to post before the deadline and get feedback from us! We want your projects to be as successful as possible, and having a good proposal is the first step towards that.
## Project Requirements

- Projects may be done in teams of up to 3 people; if you want to work on a larger team, ask us first
- The topic of your project should be relatively novel to you and your teammates (we want you to explore new things!)
- We expect about 3 hours of work per person per week
- Must relate to systems programming in a meaningful way (if you’re not sure about this, ask us)
Project Proposals

Proposal Components

Your proposal should include the following:

- Overview (Alternatively, “Explain it to me like I am five”)
- Purpose (Why are you doing this? Why is it interesting?)
- Expected workload distribution among group members
- Projected project milestones (What will you do in the first month? Second month? etc.)
- Challenges you expect will arise in working on this project