High-level Objectives

• Learn how an OS works
  – OS are essential to everything we do with computers

• Get an inside view ("internal perspective")
  – Look at design & implementation

• Learn by doing ("concrete approach")
  – You’ll code a substantial part of an actual OS

A bit of history

• Joined department in 2004
• Revised core OS course in 2005, taught as CS 3204 from 2005-2009
• 2009 Revision of undergrad curriculum, creation of
  – CS 3214 Computer System (required course)
  – CS 4284 S&N Capstone (elective)
Industry Feedback on old CS3204

We’ve hired a few students and interns recently from Tech. During the interview process, and during conversations with faculty and students, it became apparent that CS3204 Operating systems, and specially the project oriented version had a great impact in our interest in a candidate. We consistently could distinguish between systems oriented students/graduates applying for internships or full time positions based on how they described their experiences in that class. We also believe that they learnt a lot through this course, giving them a distinct edge over candidates from some of the other schools we recruit from, ranging from scheduling primitives and memory management to file system basics and IPC.

Class Format

- Lectures
- No exams
- No homeworks
- 4 Structured Projects
- 1 Open-ended project
  - Presentation
  - Poster

Prerequisites

- Willingness to master an intellectually challenging subject
- Solid knowledge of C
- Formal prerequisite: C or better in CS 3214
  - Informal prereq: B or better

Programming Projects

- 4 Projects: 1, 2, 3, and 4.
- Projects 1-4 will use the Pintos operating system developed at Stanford.
- Project dependency graph

Project Difference #1

- The Pintos projects have reputation for being hard
  - But you’ve been through CS 3214!
- Expect a stimulating challenge
- About 2,500 lines of code (over the entire semester) to write
- We will help:
  - Provide tools, answer questions
  - Although Pintos is a true OS, we run it in a simulator or emulator (Bochs or Qemu) – reproducibility!
  - Also possible: Run Pintos on actual hardware!
- Ask lots of questions!

Project Difference #2

- The projects are group projects
- Working in a group more closely resembles what you do outside of academia
  - Can design together, code together
  - Practice group collaboration tools
- Group members must contribute equally
- 2-3 students per group
  - All project-related policies are identical.
Project Difference #3

• Read before you code
• We can’t build an OS from scratch in a semester
  – Start with very primitive baseline code
• You must read a substantial amount of (well-written, well-documented) code before starting the projects
  – All of Pintos is about 7,000 lines
  – Must intimately understand probably 500-1000 lines, will be introduced gradually
  – Welcome to read all nitty-gritty details

Open Ended Project

• Extend Pintos in an interesting way!
  – Multi-threading
  – Networking
  – IPC
  – Shared memory
  – Sbrk()
  – CFS
  – …

Project Difference #4

• Only 75% of your grade comes from test cases
  – All test cases are public
• 25% is given for design & documentation
  – Will grade both design and code quality
  – Explain your design rationale
    • Better designs will yield generally more credit
    • Flawed design will incur deductions
    • Bugs that become apparent from your description will lead to deduction
  – Create maintainable code, of “peer review” quality
  – Don’t be fooled: this doesn’t mean your implementation counts for only 75% - we don’t give credit for descriptions of unimplemented designs, or designs where tests show you didn’t implement them correctly

Late Policy

• No late submissions will be accepted.
• Instead, you have 4 late days for projects 1-4:
  – Self-granted extensions, no need to ask for permission
• Contact instructor in extraordinary circumstances only
  – Job interviews do not count

Grading

• Tentative breakdown (subject to change):
  – 50% Projects 1-4
  – 20% Poster & Presentation
  – 30% Open Ended Project
• Not bound to any curve in this capstone class
  – Grades will depend on performance
  – Good performance -> good grade
• “Auto-Fail Rules”, borrowed from old CS 3204
  – Must pass 90% of tests included in Project 2 by end of semester
  – Must show “strong effort” in open-ended project

Forum Rules (aka Lex skottie)

• Not allowed:
  – Posting of any code that is part of your solution to the forum (*)
  – Posting the answers to design document questions
  – Posting detailed descriptions of your group’s design
  – Uncivil behavior
• (*) Exception: “1-line-rule”
  – Can post 1 line if it causes a compile-time error
• Ok to post:
  – Backtraces, debugging output, debugger messages
  – Illustrating example for a technique
  – Questions & explanations relating to concept
  – Questions & answers relating to Pintos in general
• If in doubt, ask before posting!
Honor Code

- Will be strictly enforced in this class
  - Will not give warning or engage in discussions before filing honor code cases where I believe they are warranted
- Do not cheat
  - Observe collaboration policy outlined in syllabus
- Will use MOSS for software cheating detection
  - Do not borrow code from other offerings
  - Follow collaboration policy
- Read all policies posted on the website
  - “I was not aware…” is no excuse
- If in doubt, ask!

Acknowledgements

- Will draw in lectures from
  - Silberschatz et al’s book (“Dinosaur book”)
  - And other texts, in particular Stalling’s book and Tannenbaum’s Modern Operating Systems
  - Course material created in other courses using Pintos
    - E.g., CS140 @ Stanford, CS 326 @ U San Francisco
  - Course material created by McQuain & other VT instructors
  - And other sources

Reading Material

- Optional Textbook