CS 3204
Operating Systems

Lecture 1

Godmar Back

Virginia Tech

Why study operating systems?

Virginia Tech

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

About Me

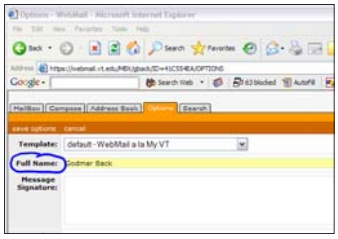
- Undergraduate Work at Humboldt and Technical University Berlin
- PhD University of Utah
- Postdoctoral Work at Stanford University
- 5rd Year at Virginia Tech as Assistant Professor (joined August 2004)
  - 5th time teaching this class
- Research Interests:
  - Operating systems, runtime systems and compilers: focus on building reliable systems.

Course Facts

- Meet Tuesday & Thursday 9:30am-10:45am (MCB 231) or 11:00am-12:15pm (MCB 230)
  - Will try to keep the two sections in sync
- Check website regularly
  - http://courses.cs.vt.edu/~cs3204/fall2008/gback
- Use CS Forum for projects first
- Second - send project-related email to
  - cs3204-staff@cs.vt.edu
- TAs:
  - Peter Radics (100%)
  - Xiaomo Liu (50%)

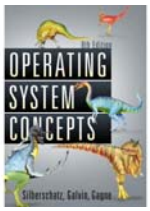
Email Etiquette

- Please enter your name in webmail so it appears in From: line
- Be coherent when you email



Reading Material

- Required Textbook
  - Silberschatz, Galvin, Gagne: *Operating Systems Concepts (8th Edition), 2008*
  - 7th may be ok
  - Will post reading assignments:
    - Chapter 1 & 2 for this week



Class Format

- Lectures
- Exams
  - 1 Midterm: Tentative Date is Thursday, Oct 2
  - 1 Final (Comprehensive)

  Exams are only offered at the announced time. Missed exams result in zero score.
- Programming Projects

Prerequisites

- Willingness to master an intellectually challenging subject
- Knowledge of computer organization (CS/ECE 2504)
  - Addressing, Registers, Basic Assembly Language, Memory Organization
- Knowledge of algorithms & data structures (CS 2604)
  - Solid knowledge of linked lists, hash tables, etc.
- Solid knowledge of C

Talking about C…

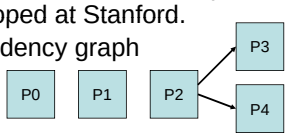
```
#define offsetof(TYPE, MEMBER) \
        ((size_t) &((TYPE *) 0)->MEMBER)
struct point {
   int x;
   int y;
   int z;
   char c;
   float o;
};
```

0: x (4 byte int)
4: y (4 byte int)
8: z (4 byte int)
12: c (1 byte char)
13: 3 byte padding
16: o (4 byte float)

Q.: What is offsetof(struct point, y)?
offsetof(struct point, o)?

Programming Projects

- 5 Projects: 0, 1, 2, 3, and 4.
- Project 0 is warm-up
- Projects 1-4 will use the Pintos operating system developed at Stanford.
- Project dependency graph



- Projects are different in five ways

Project Difference #1

- The Pintos projects are hard (*)
- Expect a stimulating chal
- About 2,500 lines of code 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!
  - New this semester: Run Pintos on actual hardware!
- Ask lots of questions!

* but - in my opinion - not unreasonably hard and they compare well to what our peer institutions do – recall that CoE was just ranked #14 among undergraduate Eng colleges!)

Industry Feedback on 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.

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
  - Learn group collaboration tools (CVS version control system)
- Group members must contribute equally
- 3 People per group, this semester groups can be formed across sections.
  - 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

Project Difference #4

- Only 50% of your grade comes from test cases
  - All test cases are public
- 50% is given for design & documentation
  - Will grade both design and code quality
  - Requires design documents
    - For P1 & P2, create these individually
    - For P3 & P4, one document per group suffices.
  - 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 50% - we don't give credit for descriptions of unimplemented designs, or designs where tests show you didn't implement them correctly

Project Difference #5

- We use C, not C++
- Note: C is by and large a subset of C++
- Don't have virtual methods, don't have templates, don't have rtti
- Still use object-orientation, still use encapsulation
- Most OS are written in C, not C++.
- C is the predominant language in embedded systems

Late Policy

- No late submissions will be accepted.
- Instead, you have 4 late days:
  - 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):
  - 15% Midterm
  - 30% Final
  - 55% Projects
- Not grading on a standard scale; grade will be based on a curve:
  - Median will divide B- and B
  - Grading on a curve means every assignment is important
- Additional stipulations to pass the class (aka "Auto-Fail Rules")
  - Must pass 90% of tests included in Project 2 by end of semester
  - Must show "reasonable effort" in both Project 3 & 4
  - Necessary, not sufficient conditions

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 iff 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!

Why cheating successfully is hard

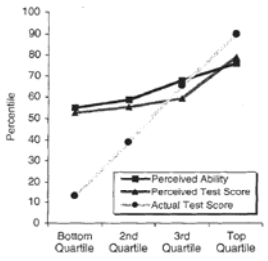


Figure 4. Perceived logical reasoning ability and test performance as a function of actual test performance (Study 4).

Source: Justin Kruger; David Dunning (1999). "Unskilled and Unaware of It: How Difficulties in Recognizing One's Own Incompetence Lead to Inflated Self-Assessments". *Journal of Personality and Social Psychology* **77** (6): 1121–34

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