CS 3204
Operating Systems

Lecture 1

Godmar Back

High-level Objectives

- Learn how an OS works
  - OS are essential to everything we do with computers
- Get an inside view
  - Look at design & implementation
- Learn by doing
  - You'll code a substantial part of an actual OS

About Me

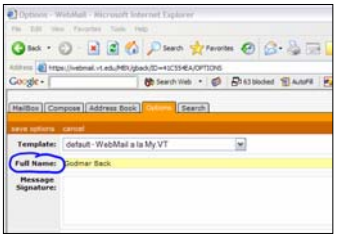
- Undergraduate Work at Humboldt and Technical University Berlin
- PhD University of Utah
- Postdoctoral Work at Stanford University
- 3rd Year at Virginia Tech as Assistant Professor (joined August 2004)
- Research Interests:
  - Operating systems, runtime systems and compilers: focus on building reliable systems.

Course Facts

- Meet Tuesday & Thursday 11:00am-12:15pm MCB 216
- Check website regularly
  - http://courses.cs.vt.edu/~cs3204/fall2006/gback
- Send class-related email to
  - cs3204-staff@cs.vt.edu
- Use CS Forum for projects
- TA: Shuaiwen Song ("Leon") for this section

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 (7th Edition), 2005*
  - Will post reading assignments:
    - Chapter 1 & 2 for this week

Class Format

- Lectures
- Exams
  - 1 Midterm
  - 1 Final (Comprehensive)

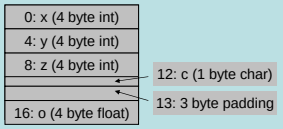
  You must take both exams to pass the class.
- Programming Projects

Prerequisites

- Willingness to master an intellectually challenging subject
- Knowledge of computer organization (ECE 2504)
  - Addresses, 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…

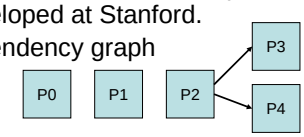
```
#define offsetof(TYPE, MEMBER) \
        ((size_t) &((TYPE *) 0)->MEMBER)
struct point {
    int x;
    int y;
    int z;
    char c;
    float o;
};
```



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 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 (Bochs)→ reproducibility! (well, almost…)
- 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
  - Learn group collaboration tools (CVS version control system)
- Group members must contribute equally

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
  - Requires design documents
  - Explain your design rationale
  - Create maintainable code, of "peer review" quality
  - Will grade on code quality
  - no credit for descriptions of unimplemented designs, though

Project Difference #5

- We use C, not C++
- Note: C is 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++.

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 all tests of Project 2 by end of semester
  - Must show "reasonable effort" in both Project 3 & 4
  - Necessary, not sufficient conditions

Honor Code

- Will be strictly enforced in this class
- 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