CSE 120
Principles of Operating Systems

Fall 2001

Lecture 1: Course Introduction

Geoffrey M. Voelker

Lecture 1 Overview

- Class overview
- What is an operating system?
- Operating system modules, interfaces, structure

CSE 120 Class Overview

- Course material taught through class lectures, textbook readings, and handouts
- Course assignments are
  - Homework questions from the book
  - Three large programming projects
- Discussion sections are a forum for asking questions
  - Primarily about lecture material and homework
  - But also “guest” discussions from the project TAs
  - Will have mailing list and online discussion forums, too

Homeworks

- There will be approximately five homeworks throughout the quarter
  - Reinforce lecture material…no better practice
- Collaboration vs. cheating
  - I encourage you to discuss homework problems with others
    - » You can learn a lot from each other
  - But there is a distinction between collaboration and cheating
  - Rule of thumb: Discuss together in library, walk home, and write up answers independently
  - Cheating is copying from other student’s homeworks or solution sets, searching for answers on the Web, etc.
  - Suspicious homeworks will be flagged for review by me

Nachos

- Nachos is an instructional operating system
  - It is a user-level operating system and a machine simulator
    - » Not unlike the Java runtime environment
    - » Will become abundantly clear (or not so clear) very soon
  - Programming environment will be C++ on Unix (Linux/Solaris)
  - The projects will require serious time commitments
    - » This is not an understatement
- You will do three projects using Nachos (more later)
  - Concurrency and synchronization
  - Multiprogramming
  - Virtual memory
- You will work in groups of four on the projects
  - Start identifying partners now

Labs

- We will have access to two labs in the AP&M basement
  - uAPE: sparc/Solaris
  - OSTL: x86/Linux (access is combo controlled)
- You can use either platform for your project
  - Or even your home machine
  - Helps if everyone in your group is using the same platform

Exams

- Midterm
  - Thursday, October 25
  - Covers first half of class
- Final
  - Friday, December 7
  - Covers second half of class plus selected material from first half
    - » I will be explicit about the material covered
- Crib sheet
  - You can bring one double-sided 8.5x11” page of notes to each exam to assist you in answering the questions
  - Note: Not a substitute for thinking

Grading

- Homeworks: 20%
  - Think of these collectively as a take-home midterm
- Midterm: 20%
- Final: 25%
- Projects: 35%

How *Not* To Pass CSE 120

- Do not come to lecture
  - It's too early, the slides are online, and the material is in the book anyway
  - Lecture material is the basis for exams and directly relates to the projects
- Do not do the homework
  - It's only 20% of the grade
  - Excellent practice for the exams, and some homework problems are exercises for helping with the project
  - 20% is actually a significant fraction of your grade (difference between an A and a C)

How *Not* To Pass (2)

- Do not ask questions in lecture, office hours, or email
  - It's scary, I don't want to embarrass myself
  - Asking questions is the best way to clarify lecture material at the time it is being presented
  - Office hours and email will help with homeworks, projects
- Wait until the last couple of days to start a project
  - We'll have to do the crunch anyways, why do it early?
  - The projects cannot be done in the last couple of days
  - Some groups last time learned that starting early meant finishing all of the project on time…and some didn't

CSE 120 Course Material

- This course addresses classic OS concepts
  - The services provided by the OS
  - OS implementation on modern hardware
  - Co-evolution of hardware and software
  - Techniques for implementing software systems that are
    - » Large and complex
    - » Long-lived and evolving
    - » Concurrent
    - » Performance-critical
- System software tends to be mysterious
- Our goal is to reveal all mysteries

Fundamental OS Issues

- The fundamental issues/questions in this course are:
  - Structure: how is an operating system organized?
  - Sharing: how are resources shared among users?
  - Naming: how are resources named (by users and programs)?
  - Protection: how are users/programs protected from each other?
  - Security: how can information access/flow be restricted?
  - Communication: how to exchange data?
  - Reliability and fault tolerance: how to mask failures?
  - Extensibility: how to add new features?

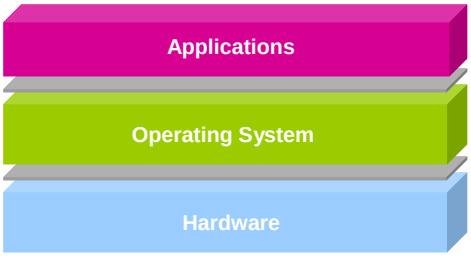
Fundamental OS Issues (2)

- Concurrency: how to control parallel activities?
- Performance: how to make efficient use of resources, reduce OS overhead?
- Scale and growth: how to handle increased demand?
- Compatibility: can we ever do anything new?
- Distribution: how to coordinate remote operations?
- Accountability: how to charge for/restrict use of resources?

- And the principles in this course are the design methods/approaches/solutions to these issues

What is an operating system?

- The operating system is the software layer between user applications and the hardware



- The OS is “all the code that you didn’t have to write” to implement your application

The OS and Hardware

- The OS abstracts/controls/mediates access to hardware resources
  - Computation (CPUs)
  - Volatile storage (memory) and persistent storage (disk, etc.)
  - Communication (network, modem, etc.)
  - Input/output devices (keyboard, display, printer, camera, etc.)
- The OS defines a set of logical resources (objects) and a set of well-defined operations on those objects (interfaces)
  - Physical resources (CPU and memory)
  - Logical resources (files, programs, names)

The OS and Hardware (2)

- Benefits to applications
  - Simpler (no tweaking device registers)
  - Device independent (all network cards look the same)
  - Portable (same program on Windows95/98/ME/NT/2000/…)
  - Transportable (same program across different OSes (Java))

The OS and Applications

- The OS defines a logical, well-defined environment…
  - Virtual machine (each program thinks it owns the computer)
- For users and programs to safely coexist, cooperate, share resources
  - Concurrent execution of multiple programs (timeslicing)
  - Communication among multiple programs (pipes, cut & paste)
  - Shared implementations of common facilities
    - » No need to implement the file system more than once
  - Mechanisms and policies to manage/share/protect resources
    - » File permissions (mechanism) and groups (policies)

OS Metaphors

- Service provider
  - The OS provides a standard set of facilities/services that enable programs to be simple and portable
- Executive/bureaucrat/big brother/juggler
  - The OS controls access to shared resources, and allocates resources for the greater good
- Caretaker
  - The OS monitors and recovers from exceptional conditions
- Cop/security guard
  - The OS mediates access to resources, granting or denying requests to use resources

For next class...

- Browse the course web
  - http://www-cse.ucsd.edu/classes/fa01/cse120/
- Read Chapters 1 and 12
- Send your email address to me (voelker@cs.ucsd.edu) for mailing list
- Start thinking about partners for project groups
- No discussion section on Monday
  - Anirban is AWOL (stuck in India)
- I will bring account slips at the next lecture (not ready before class today)