Operating Systems Design (CS 423)

Elsa L Gunter
2112 SC, UIUC
http://www.cs.illinois.edu/class/cs423/

Based on slides by Sam King and Andrew S Tanenbaum

Contact Information - Elsa L Gunter

- Office: 2112 SC
- Office hours:
  - Mondays, Wednesdays 11:00am - 12:00pm, Fridays 9:00am – 9:50am
  - May change if these prove inopportune
  - Also by appointment
- Email: egunter@illinois.edu

Contact Information - TAs

- Teaching Assistants Office: 0207 SC
- Nipun Sehrawat
  - Email: sehrawa2@illinois.edu
  - Hours: Not yet set

Course Website

- Main page - summary of news items
- Policy - rules governing course
- Lectures - syllabus and slides
- MPs - information about homework
- Exams
- Unit Projects - for 4 credit students
- Resources - tools and helpful info
- FAQ

Some Course References

- No required textbook.
- *Modern Operating Systems (third edition)*, by Andrew S Tanenbaum, published by Peason/ Prentice Hall
- *Virtual Machines – Versatile Platforms for Systems and Processes*, by James E. Smith and Ravi Nair, published by Elsevier/Morgan Kaufmann
- (Optional) *Understanding the Linux Kernel*, by Daniel P. Bovet and Maroc Cesati, published by O'Reily

Course Grading - MPs

- Homework 20%
  - Traditionally 4 MPs.
    - More hard than long
    - I will try to break them up into more smaller ones.
  - MPs submitted by **handin** on EWS linux machines
  - Late submission penalty: 20% of assignments total value

Course Grading - Exams

- 2 Midterms - 20% each
  - In class – **Mar 2, Apr 18**
- **DO NOT MISS EXAM DATES!**
- Final 40% - May 6, 8:00am – 11:00am
- Percentages are approximate
  - Exams may weigh more if homework is much better
- No extra credit
- If final is better than average of other scores, drop lowest score, make final 60%

Default Course Grade Scale

A+	≥ 97	C+	≥ 77
A	≥ 93	C	≥ 73
A-	≥ 90	C-	≥ 70
B+	≥ 87	D	≥ 60
B	≥ 83	F	< 60
B-	≥ 80		

May curve if scores differ much from previous semesters

Course Homework

- You may discuss homeworks and their solutions with others
- You may work in groups, but you must list members with whom you worked if you share solutions or solution outlines
- Each student must turn in their own solution separately
- Read http://www.cs.uiuc.edu/class/sp11/cs423/policy.html#collaboration for full statement
- You may look at examples from class and other similar examples from any source
  - Note: University policy on plagiarism still holds - cite your sources if you are not the sole author of your solution
- Problems from homework may appear verbatim, or with some modification on exams

Prerequisites

- CS241 or ECE391–Important
- You must be comfortable with C/C++
- You must know basics about systems
  - E.g., you should know what a file descriptor is
- Will give a brief review next class

Facilites

- Lab: EWS Machines
  - ssh –Y <*netid*>@remlnx.ews.illininois.edu
- Virtual Machine Emulator
  - Qemu: http://wiki.qemu.org/Main_Page
  - Executable available from ~cs423/bin
- Will use Linux Kernel 2.6.37
  - http://www.kernel.org/

Course Objectives

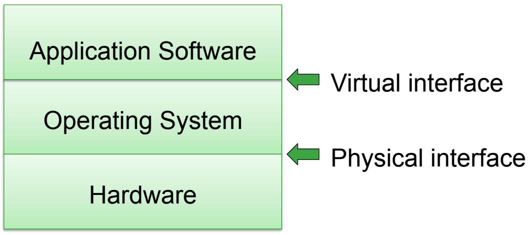
- Understanding Major Concepts of Operating Systems
  - Threads and synchronization
  - Virtualization and Virtual Machine Monitors
  - I/O and Device Drivers
  - File Systems
  - Distributed Systems
  - Security
- Understanding of Interaction Between Hardware and Kernel
- Understanding of the Layers of Abstraction and Virtualization of Operating Systems

What is an Operating System?

- Most basically, what is an operating system are the main things it does?

What is an Operating System?

- Software layer between hardware and application software



What Does an Operating System Do?

- Abstraction
  - Presents application with uniform, (relatively) simple access to resources
- Regulation
  - Governs access to resources to guarantee proper use

What Does an Operating System Do?

- In any OS area, ask:
  - What resource(s) is involved?
  - What is physical reality of resource?
  - What abstraction to present to user apps?
  - What protections to guarantee?

Dual Relationship Between OS and Apps

- User app main program; calls kernel for services

- OS main program; calls user programs as subroutines

Main Roles of OS

- Illiusionist
  - Make resources seem better than they are
    - Examples?
- Government
  - Parcel out shared resources to multiple apps in a fair, safe, efficient way
    - Tax: Costs CPU, Memory,
  - Maintains Protections
    - Allows good programs to just mind their own business

Why Study Operating Systems?

- Some day, you may write one (or part of one)
- OS concepts ant techniques common in other domains
  - OS is huge multi-threaded, event drive app
  - Designing and implementing abstractions and protections
  - Caching, indirection, concurrency, atomicity, …
- Fun to “open the hood and look inside”