Operating Systems I

COMS W4118

Prof. Kaustubh R. Joshi

krj@cs.columbia.edu

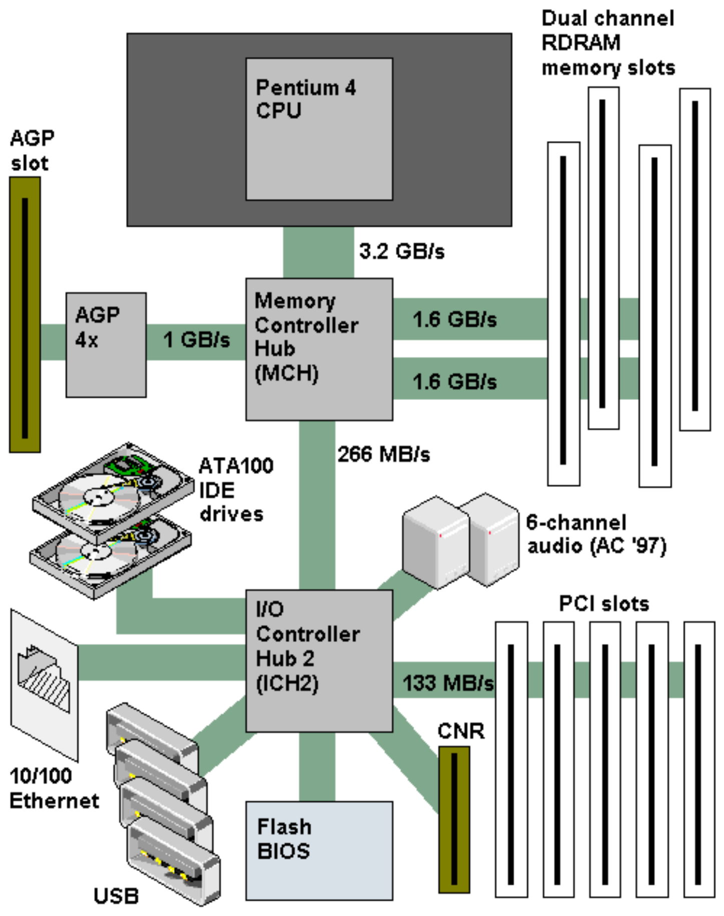
http://www.cs.columbia.edu/~krj/os

References: Operating Systems Concepts (9e), Linux Kernel Development, previous W4118s

What is an OS?

Software that converts this:

From Computer Desktop Encyclopedia
© 2001 The Computer Language Co. Inc.



into this:

What does an OS do?

- Government
  - Always running
  - Does the things programs can't do for themselves
- Traffic cop
  - Arbitrate resource usage between competing interests
- Security guard
  - Prevents bad actors from harming others
- Assistant
  - Provides services and abstractions to hide complexity and make programmer's life easier
- The face of the machine
  - Facilitates interface to the user

What does an OS do?

- Support multiprogramming
- Resource allocation
- Isolation
- Abstraction
- Shared facilities and libraries

Resources

- An OS usually multiplexes at-least the following resources
  - CPU
  - Memory
  - Disk space
  - I/O bandwidth
  - Network bandwidth
  - Access to devices
    - Display, Keyboard, Audio
- In this class, we will learn how

Resource Allocation

- We will learn how access to resources is arbitrated through the following OS functions
  - CPU scheduling
  - Memory management
  - Disk scheduling
  - I/O scheduling
- Strive to share fairly among multiple users
  - But what's a user?
  - And what's fair?

Isolation

- Prevent one user from interfering with another user's data
  - Other users' files
  - Other users' traffic
  - Other users' private data structures
- Prevent users from monopolizing resources
  - Starve other users from CPU or bandwidth
  - Denial of service
- Protect the OS itself
  - From attempts to subvert its isolation mechanisms

OS Abstractions

- And finally, we will learn about the following OS abstractions
  - Processes
  - Virtual memory
  - Threads
  - Locks
  - File systems
  - Communication channels

That’s it? What about xxx?

- Modern OSes are big. They provide many services. E.g., a window manager, compilers, media players, browsers
- But what’s really part of the OS?
  - No single answer, even in the same OS family
  - Okay, then what’s the minimal common subset?
- The kernel
  - Core piece that runs with special privileges
  - Enables higher level services
  - We’ll focus on that in this course

How will we learn about OSes?

- General concepts
  - OS principles from the textbook
  - Evaluated through homeworks and quizzes
- Hands on experience
  - Android Mobile OS
  - Based on the Linux kernel
  - Code that runs on 100s of millions of phones
  - We'll run it in a full system emulator
  - Programming assignments change the kernel
  - Add new functionality

About me

- Dr. Kaustubh Joshi (KJ)
  – Pronounced "cow" + "stub"
  – Forget the 'h'!
- I'm an adjunct professor
  – Which means I have a day job
  – My availability in the department is extremely limited
  – Office hours only
- Principal researcher at AT&T Shannon Labs
  – Florham Park, NJ and downtown NYC
  – Work on dependable large scale distributed systems
  – Data centers, the phone network
  – Currently, work on future cellular network architectures
  – How phone OSes and networks can co-operate better

Caveat Emptor!

- I don't know the usual workload of your other classes
  - Tried to keep the structure similar to previous years
- **This** class is intense
  - You will be doing a **lot** of coding
  - Working on kernels is hard and time consuming
  - Do **not** take it if you aren't really interested
  - I've been told there are alternatives
- But … hopefully rewarding
  - Real kernel hacking
  - Work with Android
  - A feeling of accomplishment
  - Valuable job skills

Prerequisites

- W3827: Computer Architecture
- W3137: Data structures and algorithms
- Unix toolchains: diff, make, gcc, etc.
- **Can program in C, shell scripts**
- **Aren't intimidated by a lot of (others) code**
- We're working with the Linux kernel
- No Java, Python, etc.
- If you don't know C, don't take the class
- Finally, lots of time

Logistics

- Class website
  - http://www.cs.columbia.edu/~krj/os
  - Website is your first stop for everything
  - Will be using Courseworks only for grades and possibly coordinating demos
- Class Forum
  - Through Piazza
  - https://piazza.com/class#spring2013/comsw4118
  - If you haven't been signed up already, do so now
  - All questions should be directed through the forum
  - Use forum to self-organize into project groups (later)

Personnel

- Instructor:
  - Office hours: Mon 4-5pm, Adjunct office CS457
  - Contact information on class site
  - Talk to TAs first, get in touch with me after
- TAs:
  - Angela Wei, acw2163
  - Jiao Li, jl3931
  - More coming soon…
  - Check website for office hours

Required Textbooks

- ***Operating System Concepts***, 9th Edition by Abraham Silberschatz, Peter B. Galvin, Greg Gagne. ISBN 978-1-1180-6333-0. December 2012.
- ***Linux Kernel Development***, 3rd Edition by Robert Love. ISBN: 978-0672329463. July 2010.
  - Both books available via Amazon
  - Rental or sale. Kindle editions are also available
  - Make sure you have the right edition
  - Written homeworks will reference problems from the Concepts book

Optional Reference

- ***Understanding the Linux Kernel***, Third Edition by Daniel P. Bovet and Marco Cesati Ph.D. ISBN 978-0596005658. November 2005.
  - Older book with some kernel details out of date
  - But, a lot more detail than LKD

Lectures

- Course outline and reading assignments from Concepts and LKD are on the course website.
- Will upload slides to course website **after** class
  - I don't want laptops and distractions
  - I don't want peeking
  - I realize you may want to take notes on slides
  - This may change depending on class dynamics…
- Reading assignments and topic ordering may change. Check website often.

Grading

- 50% Assignments
- 20% Midterm
- 30% Final
- I have no idea what to expect
- So, you should expect a curve
- It’s a intense course – I won’t be brutal
- Regrading requests to TAs within two weeks of grade being made available. Contact me only if you can’t resolve the matter with the TA.

Homework

- Lots of it…
- 5 assignments
- Count for 50% of your grades
- Each assignment will have a written and a programming part
- All assignments count towards grade

Written Assignments

- Usually count for 40% of each homework
- To be done individually
- Submit electronically by 12:01AM the day of the due date, i.e., midnight before class day
- Homework received later that day loses 10% credit, thereafter zero credit
- No exceptions other than a letter from dean or a doctor's note

Programming Assignments

- Each homework will have a programming assignment
- Usually worth 60% of the grade
- First assignment is individual
- Thereafter, in groups of 3
- Form your groups by Feb 4 using the
- Thereafter, I will fill in the blanks

Programming Environment

- You will need a CLIC account
- First assignment to be done on Unix environment
- Subsequent assignments involve modifying the Android kernel
- Testing will be done using the Android emulator
- We will provide environment with the appropriate tools
- Grading will be done on CLIC machines
- If your code doesn't compile or run there, expect a zero
- You can also use your own laptops – your own on your own if you run into trouble, but we will provide pointers to help you get started

Working with Kernels

- Working with kernels is hard
- Long code/test cycle. Have to wait to boot. Android emulator is especially slow
- Hard to debug
- May feel like pulling your hair out at times…
- …remember, it doesn't grow back
- Compile incrementally
- printk is your friend
- Using the emulator helps

The Android Emulator

- We'll run it in a virtual machine
- The emulator itself implements the entire processor in software using qemu
- This means you can use GDB

Unsupported but recommended

- If you feel adventurous…

- Get an Android Nexus tablet!
- http://www.google.com/nexus/7/
- Reasonably priced - $199 onwards
- Put your code on it. You'll learn a lot

Exams

- Both exams will be closed book, laptop, phone, tablet, etc.
- Old fashioned calculators are okay
- No calculator apps

Honesty

- Read the class website honesty policy
  - You know all this already
- General discussion (through forum) is good
- Your work must be your own, or your group's, in the case of group assignments
  - Don't use answers or code you got from your friends or found on the internet
  - Explicitly cite all your sources
- There will be zero tolerance for cheating
  - First offense gets a significant grade downgrade
  - Repeaters or egregious offenses will fail the class and be reported for further disciplinary action

Expectations

- You're all adults
  - We'll make the homeworks as detailed as possible
  - But we won't spell out all the details
  - You'll need to fill in the blanks
  - And make sound engineering decisions
  - You'll be evaluated on that
- I'm new at this
  - If something is missing I won't know unless you tell me
  - If you notice any mistakes, speak up
  - If you have any suggestions or feedback about what would make the class better, get in touch

For your viewing pleasure…

- Homework #1 out on website today
- You'll be programming in a Unix environment
- You'll be writing a simple shell
- We'll be testing your programming skills
- Get a CS account (http://crf.cs.columbia.edu) ASAP if you don't have one already
- Not having an account isn't an excuse for late submission
- Have fun…