Operating Systems
ECE344

Introduction

Ashvin Goel
ECE
University of Toronto
About the Instructor

- Instructor – Ashvin Goel

- Contact information
  - Email: ashvin@eecg.toronto.edu
    - Good way to contact me directly
  - Webpage: http://www.eecg.toronto.edu/~ashvin
  - Office: SF2001B
  - Office Hour: Thu, 1-2pm

- Research Interests
  - Operating systems, storage systems, cloud systems
What is an Operating System?
What is an Operating System?

- Layer of **software** between hardware and applications
  - Application is usually dedicated to a single task
  - OS serves all applications, is **systems software**
What does an Operating System do?

- OS manages h/w (e.g., CPU, memory, disk…)
  - Provides simpler interface to devices
    - E.g., access disk as files
    - What problems occur otherwise?
  - Allows running multiple programs at the same time
    - Allows programs to share CPU, memory
    - Isolates applications from each other
      - What problems occur otherwise?
    - Isolates itself from applications
      - What problems occur otherwise?
Aren’t Operating Systems Dead?

- The OS I use has already been written, I doubt it will be my job to write another one
  - Haven’t OS developers figured everything already?
  - What more is there to do?

- Why is this a core course?
  - OS sounds cool
  - Department doesn’t know better
  - Prof wants to keep his job
  - I have no idea
Why Study Operating Systems?

- Studying OS design IS studying design of large software systems … its fun too?
- OS hacking will make you a better programmer/thinker
  - OS is really large (Windows is 50 million lines+)
  - OS manages concurrency
    - Concurrency leads to interesting programming challenges
    - Interesting challenges can lead to deep insights
  - OS manages raw hardware
    - Programming raw hardware is challenging: timing dependent behavior, undocumented behavior, h/w bugs
  - OS must be efficient
    - Applications can use resources maximally
  - OS fails \(\Rightarrow\) machine fails
    - Basis of system security, helps focus on reliability, availability
Why Won’t Operating Systems Go Away?

- Every device runs an OS
  - With IoT, every device manufacturer needs an OS

- Consider a portable music player
  - Involves many OS issues: Implements file system for storing music files, reads music from disc and buffering it, controls volume, display, wireless network access, etc.

- OS ideas applicable to all large application software
  - Virtualization software (e.g., vmware, kvm), manages virtual machine “hardware”
  - Cluster computing software (e.g., Hadoop) manages cluster
  - Cloud software (e.g., Amazon S3, Google drive) manages geographically distributed compute & storage hardware
  - Web browser manages what?
The Gears guys were thinking about a multi-threaded browser, but that led us to talk about, well, instead of multiple threads.

-- What if we have multiple processes? Each having its own memory and its own copy of the global data structures.

Chrome Process Manager

We're applying the same kind of process isolation you find in modern operating systems.

So, separate processes rendering separate tabs.
Goals of the Course

- Learn about operating systems
  - Understand the principles behind the design of an OS
    - Understand the requirements, and challenges
    - Space of possible solutions
- Build simple software that helps clarify OS concepts
  - Real systems are more difficult to build than to explain
  - The devil is often in the details
- Other benefits
  - Gain significant implementation and debugging experience
  - Learn to use software management tools
Course Contents

- Introduction to OS concepts
- Threads and processes
- Synchronization and concurrency
- Memory management, virtual memory
- Disk management and file systems
Lab Assignments

- There will be 6 labs in the course
- Labs will focus on learning OS concepts
  - Lab 1: Review of C, simple data structures, file copy program
  - Lab 2: Cooperative Threads
  - Lab 3: Preemptive Threads (builds on Lab 2)
  - Lab 4: Concurrent Web Server
  - Lab 5: Caching Web Server (builds on Lab 4)
  - Lab 6: File Systems
What to Expect From Lab Assignments

- Expect to spend significant time on lab assignments
  - The labs give specifications, not implementations
  - Allows for imagination
  - Allows for errors and frustration
  - It will be good to think of your design, before you code
  - Assume that you will do the design/coding outside lab hours
  - Use the lab to ask questions from TAs

- You will work individually for the lab assignments
  - Can’t complain about partner’s inactivity!
Suggested Textbooks

Modern Operating Systems, 4nd Edition
A. Tannenbaum

Operating Systems: Three Easy Pieces
Remzi and Andrea Arpaci-Dusseau
http://pages.cs.wisc.edu/~remzi/OSTEP/

Principles of Computer System Design
J. Saltzer, M. Frans Kaashoek
Class Resources

- Class web site available from instructor’s home page
  - http://www.eecg.toronto.edu/~ashvin/courses/ece344/current
  - Provides course outline, instructor’s lecture notes, lab info

- Piazza web site
  - Announcements
  - Discussion forum

- UG machines
  - Code for labs available under /cad2/ece344f/src
  - Past exams available under /cad2/ece344f/exams

- UoT Quercus
  - Quizzes, grades, course evaluations
Late Enrollment

- Make sure to sign up on Piazza
- Check that you have a **user number** on the UG machines
  - More details under lab assignments on web site
Grading

- Labs - 30%
- Quizzes - 30%
  - Three quizzes, 10% each
  - Details will be announced in class
- Final - 40%

Policies
- No extensions to deadlines
- Questions should be addressed on Piazza web site
- Read course webpage
Ethics and Engineering

- From the Academic Integrity Policy document:

- What if I didn't know I was cheating?
  - “You may be sanctioned for having committed an offence not only if you have intentionally committed it, but also if you *ought* to have known that you were committing an offence. Ignorance of the rules does not excuse cheating.”
Invalid Reasons

- I gave the other group my code because they really needed help and I was sick at home. I told them not to copy my code.
  - You *ought to have known* that they could commit an offense, and thus you would be a party to the offense.

- I left my USB stick in the lab computers and went to the washroom. The other group must have taken the stick, copied code, and returned the stick to the computer before I came back.
  - You *ought to have known* that someone could take your code, and thus you could be a party to the offense.

- I was working all night before the lab deadline, and my code was not working fully, and I was so tired that I lost my judgment, and ended up copying code.

- I heard that other students had raised their mark on the tester by copying code, or by subverting the marker, and I was really mad that others would get a free mark. I had worked very hard and I really wanted to make my work count.

- I copied code from the Internet and I did clearly state in my code that the code was copied.
About the Instructor

- Instructor – Ashvin Goel

- Contact information
  - Email: ashvin@eecg.toronto.edu
    - Good way to contact me
  - Webpage: http://www.eecg.toronto.edu/~ashvin
  - Office: SF2001B
  - Office Hour: Thu, 1-2pm