

Operating Systems

CSCI 3150

Introduction

Hong Xu

2023

courtesy: Ding Yuan@Toronto, Zili Shao@CUHK

This lecture

- Course information
- Why learn OS?
- What is an OS? What does it do?
- Summary

Why learn OS?

- Fulfill requirement?
- Operating System training is important
 - http://www.youtube.com/watch?v=-3Rt2_9d7Jg
 - What course is this?
 - <http://matt-welsh.blogspot.ca/2010/10/in-defense-of-mark-zuckerberg.html>
 - Software companies love OS students
 - Most big software companies have systems positions
- Academic research in OS is very influential

Goals of this course

- Understand operating system concepts
- How OS works, and more importantly, *why*?
 - What are the reasons that motivated each design?
- Basis for future learning
- Get hands dirty
- *Train your problem-solving skills!*

Who am I

- Hong Xu (call me Hong, or Henry)
 - No need to “prof” me
- Research: computer networks, machine learning systems
- Brief Bio:
 - Ph.D., University of Toronto, 2013
 - B.Engr., IE, CUHK (first-class honor), 2003-2007
 - Working with some of the largest tech companies
 - Research results being transferred

Team

- Instructor:
 - Homepage: <https://henryhxu.github.io/>
- Teaching Assistants:
 - WU, Shaofeng
 - TAN, Xin
 - Luo, Qin

Prerequisite (must have)

- C programming experiences
- Familiarity with Unix
- Computer organizations (e.g., CSCI2510, CENG3420)
 - What is an Instruction (e.g., *load*, *store*)?
 - What is CPU? Memory? Registers?
 - What is Stack? Stack pointer?
 - What is Program Counter (PC)?

Course Contents

- Overview of computer hardware
- Threads and processes
- Synchronization and concurrency
- Scheduling
- Memory Management, Virtual Memory
- Disk Management and File Systems
- Cloud computing and virtualization

Course Website

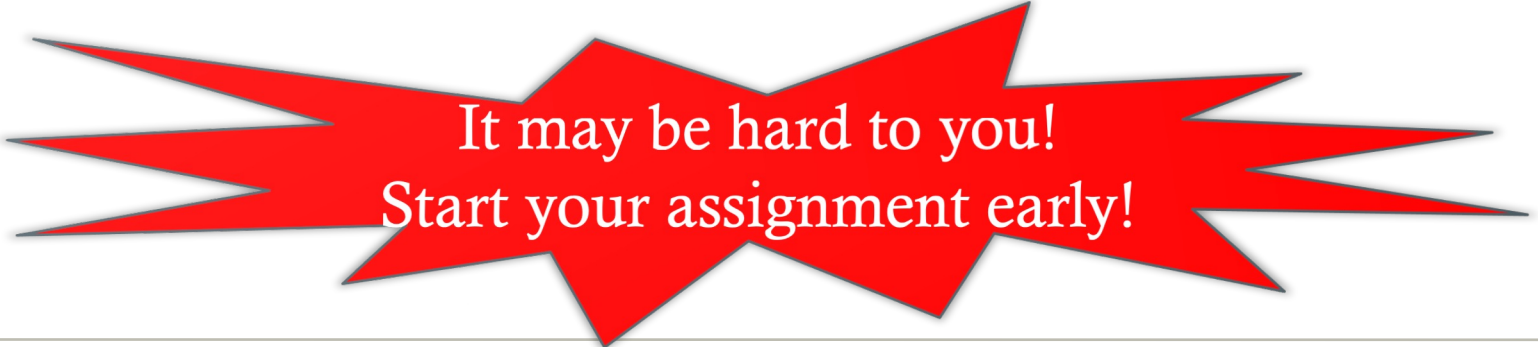
- <https://github.com/henryhxu/CSCI3150>
 - Provides slides, agenda, grading policy, etc.
 - All information regarding the labs
- Piazza used for discussion
 - <https://piazza.com/cuhk.edu.hk/spring2023/csci3150/home>
 - Sign up code is “3150 rocks!”

Grading

- Lab quizzes, 10%
 - 5 random quizzes, each worth 1 mark with 1 simple question
 - 1 midterm quiz, 5%, 20-minute written quiz
- Assignments, 50%
 - 4 assignments, programming and/or concept
- Final exam, 40%

Assignments

- Involve both programming and concept/math tasks
 - Attempt all of them independently! Similar questions may appear in the final exam
 - Some assignments will have bonus questions
- Strongly encouraged to go to the labs!
 - More info and help for assignments is provided in labs

A large, red, multi-pointed starburst graphic with a black outline, centered at the bottom of the slide. It contains two lines of white text.

It may be hard to you!
Start your assignment early!

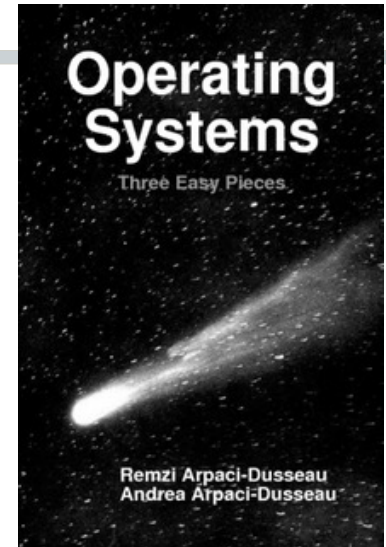
What to Expect From Assignments

- Understanding/Building an OS is difficult
- You will spend a lot of time on the assignments
 - The labs give specifications, not implementations
 - Our instructions ask that you design well, before you code
 - Assume that you will do the design/coding outside lab hours

Suggested Textbooks

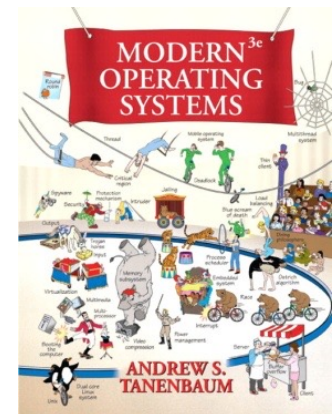
Operating Systems: Three Easy Pieces

Remzi H. Arpaci-Dusseau and Andrea C.
Arpaci-Dusseau



Modern Operating Systems, 4th Edition

Andrew S. Tanenbaum and Herbert Bos



Cheating policy

- Academic integrity
- Your work in this class must be your own – we have zero tolerance policy towards cheating of any kind and any student who cheats will get a **failure** grade in the course
- Both the cheater and the student who aided the cheater will be held responsible for the cheating

How NotTo pass

- Skip lectures
 - It's nice out there, the slides are online, and materials are in the book anyway
 - **TRUTH: Lecture material is the basis for exams**
 - It is much more efficient to learn through discussion
- Copy other people's code
 - It is cheating!

How NotTo pass (2)

- Not asking questions in lecture, office hours, or piazza
 - “I don’t want to embarrass myself”
 - TRUTH: I have very bad memory, and too many students to teach...
 - TRUTH: asking questions is the best way to clarify the material at the time it is being presented
 - *“There is no such things as stupid question...”*
- Wait until the last couple of days to start the assignment

Before we start

- Any questions?
- Though this is a hard class, OS is a very fun topic!
- Remember, we are here to help, we want you to enjoy the course (while also being able to learn something)

What is an OS?

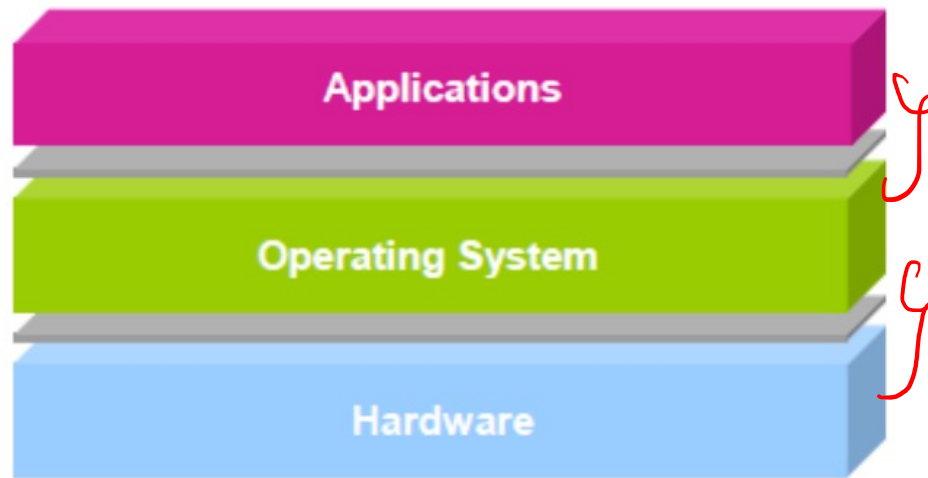
- Anyone?
- Give a few names of an OS?
 - Desktops?
 - Smart phones?

What is an OS?

- “Code” that:
 - Sits between programs & hardware
 - Sits between different programs
 - Sits between different users
- But what does it do?
 - Managing the hardware resource
 - Provide a clean set of interface to programs
- Real life analogy?
 - **Government**

OS is...

- *Software* layer between **hardware** and **applications**



- The OS is “all the code that you didn’t have to write” to implement your application
- Or, OS is a piece of software that you shouldn’t notice its existence, but you’ll feel the pain if it disappears or goes wrong

An example comparing life with/without OS

Life with an OS

```
file = open ("test.txt",  
            O_WRONLY);  
  
write (file, "test", 4);  
  
close (file);
```

Life *without* an OS

- Where is this file on disk? Which block? Which platter, track, and sectors?
- Code needs to change on a different system



OS and hardware

- The OS abstracts/controls/mediates access to hardware resources (what resources?)
 - Computation (CPUs)
 - Volatile storage (memory) and persistent storage (disk, etc.)
 - Communication (network, modem, etc.)
 - Input/output devices (keyboard, display, printer, etc.)

Benefits to Applications

- Simpler
 - no tweaking device registers
- Device independent
 - all disks look the same
- Portable
 - same program runs on
Windows95/98/ME/NT/2000/XP/Vista/7/8/10
- Worry less about interference from other applications

What does an OS do?

- Manage Resources
 - Allocation
 - Protection
 - Reclamation
 - Virtualization
- Among many running programs

What does an OS do?

- Resources
 - **Allocation**
 - Protection
 - Reclamation
 - Virtualization

- Finite resources
- Competing demands
- Examples:
 - CPU
 - Memory
 - Disk
 - Network

Government:
Limited budget,
Land,
Natural resources

What does an OS do?

- Resources
 - Allocation

- **Protection**

- Reclamation
- Virtualization

- You can't hurt me,
I can't hurt you.

- Some degrees of
safety and security

Government:
Law and order

What does an OS do?

- Resources
 - Allocation
 - Protection
 - **Reclamation**
 - Virtualization

- The OS gives,
The OS takes away

- Sometimes involun-
tarily

Government:
Income Tax

What does an OS do?

- Resources

- Allocation
- Protection
- Reclamation
- **Virtualization**

- Illusion of infinite, private resources
 - Memory vs. disk
 - Time-shared CPU

Government:
Social welfare and insurance

Why you want to learn OS?

- Foundation to other software
 - Databases, Browsers, Computational software,
- OS is one of the hardest software piece to write & debug
 - Directly talks to hardware (very ugly interfaces)
 - Abstract into clean interfaces
 - They are BIG
 - Lines of code:
 - Windows Vista (2006): 50M (XP + 10M)
 - Linux 3.6: 15.9 M
 - Android 4.0: > 1M

Why you want to learn OS?

- Many OS concepts (e.g., protection, resource management) is needed in other places
 - E.g., browser
- OS is used everywhere
 - Your car/fridge is running Linux/Windows
 - The MTR ticketing machines run Windows

Before the next class

- Browse the course web site
- Enroll yourself on Piazza
- Contact me if you have any questions
- Let the fun begin!