Welcome to CSE 4300!
Fall 2019

Today, we will see ….

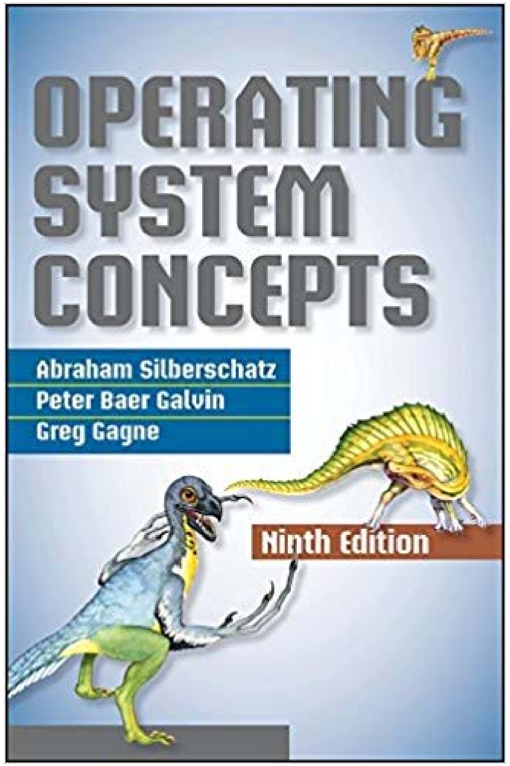
- Course information
  - Personnel
  - Policy
  - Schedule
  - Grading
- Course overview

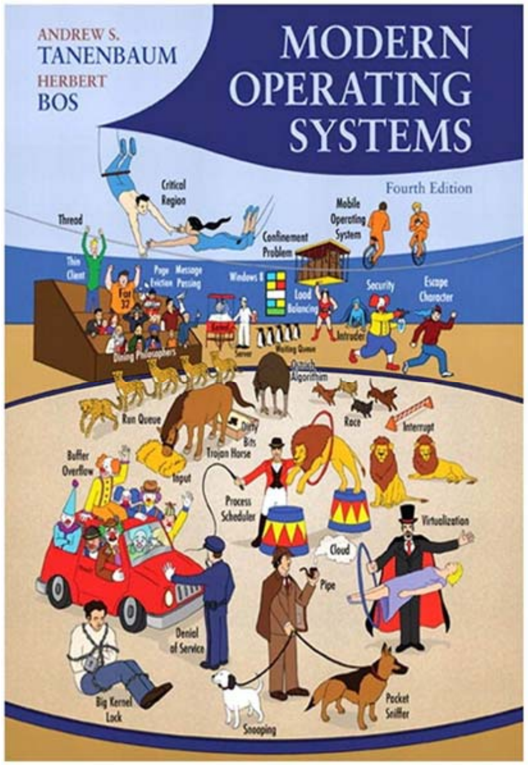
Instructor and TA

- Instructor: Song Han
- Instructor email: song.han@uconn.edu
- Office: ITEB 355
- Office hours: Tuesday 3:45-4:45pm or by appointment
- Course website
  - www.engr.uconn.edu/~song/classes/os/index.html
  - Slides and reading materials will be available on course website
  - HW/PA will be submitted through HuskyCT
- Teaching Assistant: Michelle Voong
- TA email: michelle.voong@uconn.edu
- Office: TBD
- Office Hour: TBD

Textbook

Operating System Concepts 9th Edition, by Abraham Silberschatz, Peter B. Galvin, and Greg Gagne.





In This Course …

- We will learn
  - OS design
    - Processes and threads
    - Memory management
    - File systems
    - I/O systems
    - …
  - Theories and Algorithms
  - Practice
  - Very basics of Linux
  - Systems Programming

- It will be easier for you if you have
  - Prior Experience in C programming
    - If you do not have any experience in C and you still want to take this course, I strongly suggest you to start learning ASAP

Expect …

- Fast pace
- Four programming assignments, depending on class performance/background
- Four to five homework
- One Midterm
- One Final

Grading

- Final (30%)
- Midterm (18%)
- Homework (20%)
- Programming Assignments (32%)

Submission policy

- For homework and programming assignments
  - Submit on HuskyCT and do not email the instructor/TA your homework

- Late policy and regrading
  - **20% deduction (late by 1 day), 40% deduction (late by 2days), 80% deduction (late by 3 days), and no credit if late by more than three days.**
  - Contact the instructor in case of medical emergency, and a written proof from your doctor is required.

Cheating Policy

- Zero tolerance for cheating.
- Both the cheater and the student who aided the cheater will be held responsible for the cheating.
- University policy applies for cheating or other form of misconduct.
- Punishment for cheating will be decided by the instructor as per the university guideline.

Please check the following link for important information regarding University policies on various topics (e.g., Absences from Final Examinations, Class Attendance, Credit Hour, Policy Statement People with Disabilities, Policy Against Discrimination, Harassment and Related Interpersonal Violence, the Student Code, etc.).

http://provost.uconn.edu/syllabi-references/

Please

- Attend lectures
- Submit assignments on time
- Never hesitate to ask for help
- Contact me ASAP if you have any issue

I am here for you!

Course Website Lists the Lecture Topic and Tentative Schedule

http://engr.uconn.edu/~song/classes/os/index.html

Let the journey begin!

Introduction to Operating Systems

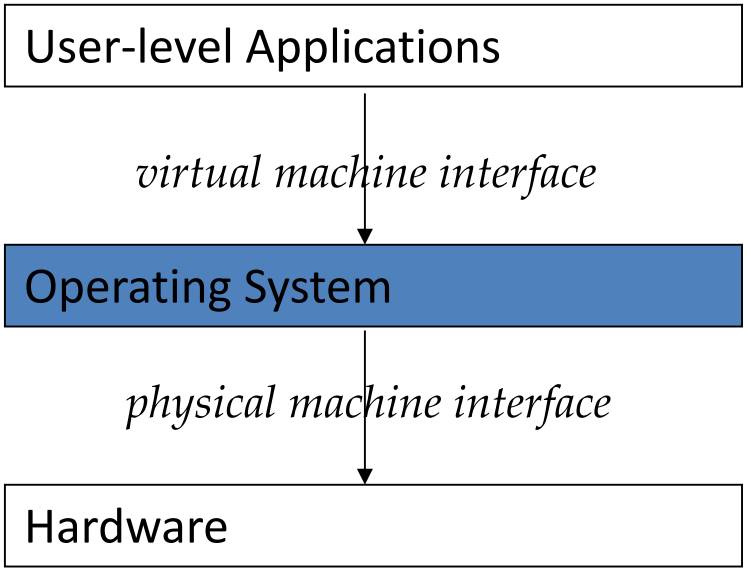
- What's an operating system (OS)?
- Why learn OS?
- Historical perspective on operating systems

OS: Examples

- MS Windows, Mac OS, Linux/Unix, Solaris, FreeBSD, Minix
- iPhone OS, Android, Symbian OS, Windows mobile, Blackberry OS, Mobile Linux, MXI (Motion eXperience Interface), embedded Linux, Windows CE
- TinyOS

OS: More Traditional View

- Interface between user and architecture
  - Hides architectural details
- Implements *virtual machine*:
  - Easier to program than raw hardware
- Provides **services** and **coordinates** machine activities



Operating Systems: Key Features

- Provides standard **services** (interface) that hardware implements
  - File system, virtual memory, networking, scheduling, time-sharing…
- **Coordinates** multiple applications and users to achieve **safety**, **fairness** and **efficiency** (high throughput)
  - Concurrency, memory protection, networking, security
- OS design challenges: **convenient** and **efficient**
  - Software engineering & systems engineering problems

Introduction to Operating Systems

- What’s an operating system? (OS)
- Why learn OS?
- Historical perspective on operating systems

Importance of Operating Systems

- **Key** component of computer systems
  - Meeting point of software & hardware
  - Understanding how computers work = understanding operating systems
    - OS provides key services required by all application programs
  - Be able to write better application-level programs!

- Rich topic:
  - OS = most complex software on your PC
    - Windows XP kernel: 40 million lines of code

New Developments in OS Design

- Operating systems: very active field of research
  - Demands on OS's growing
  - New application spaces
  - Rapidly evolving hardware (sensors, mobile devices)

- Advent of open-source operating systems – Linux
  - ➢ You can contribute to and develop OS's!
  - ➢ Excellent research platform

Introduction to Operating Systems

- What’s an operating system? (OS)
- Why learn OS?
- Historical perspective on operating systems

History of operating systems

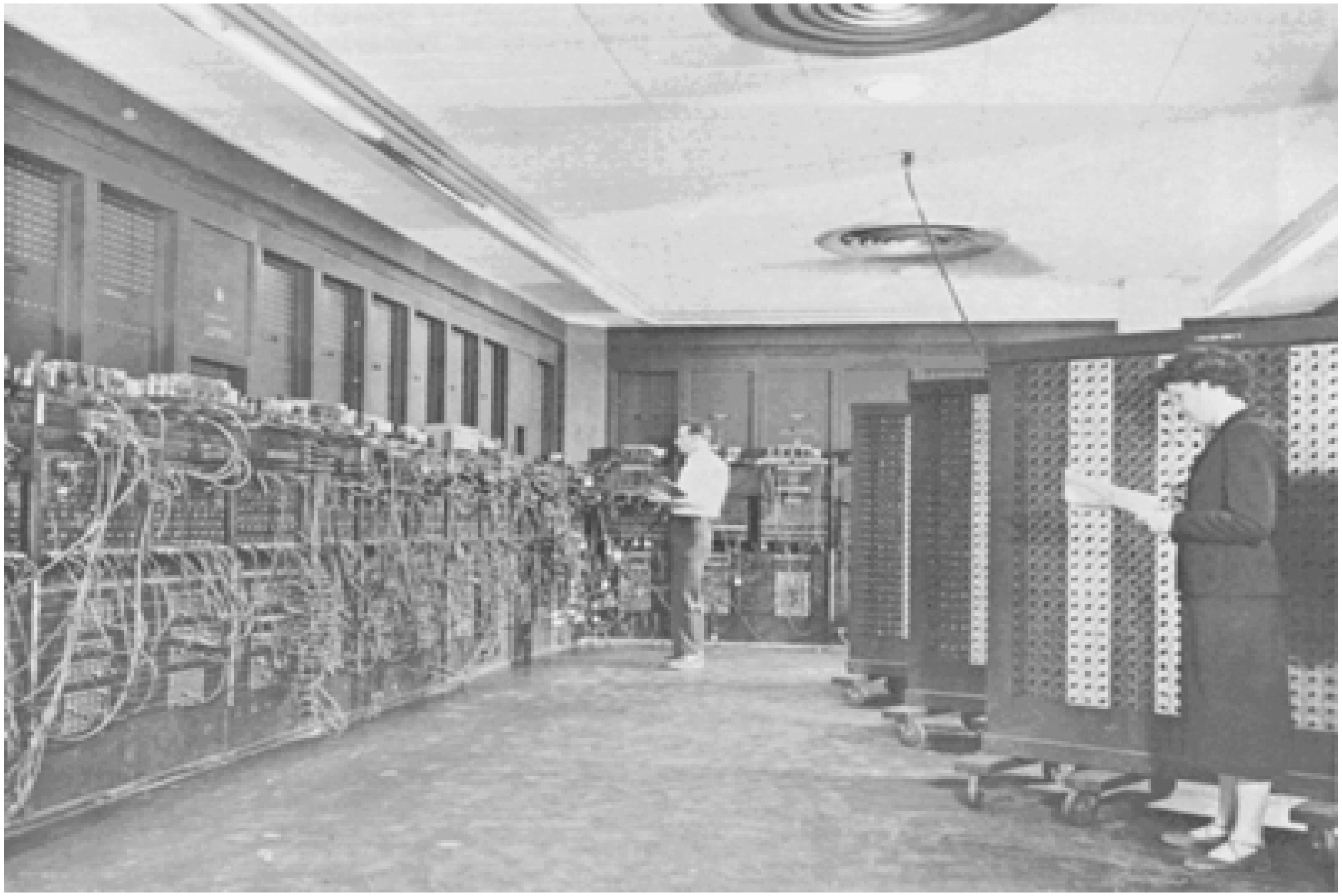
- Five waves
  - Mainframe computers
  - Minicomputers
  - Desktop
  - Internet PC
  - Mobile Internet

https://www.youtube.com/watch?v=26QPDBe-NB8

Mainframe I (1945-1955)

- Hardware technology
  - Mechanical relays, then vacuum tubes
- The Experience
  - No O.S., no library calls
  - Programming in ***machine*** language (NOT assembly)
  - Basic I/O
    - E.g. punch cards

ENIAC



Mainframe II (1955-1965)

- Hardware Technology
  - Transistors (smaller, more reliable, *affordable*)
- The Experience
  - Compilers, linkers, loaders are available!
  - Programming done in assembly and FORTRAN
  - I/O: magnetic tape
  - Batch processing

Batch Processing

- Execute multiple “jobs” in **batch**:
  - Load program
  - Run
  - Print results, dump machine state
  - Repeat
- Users submit jobs (on cards or tape)
- Human schedules jobs
- Operating system loads & runs jobs

Minicomputers (1965-1975)

- Hardware technology
  - Integrated Circuits
- The Experience
  - Multiprogramming: several programs to run at same time
  - Unix is born (1969)
    - Integration of simple tools
    - “Shell”: composable commands
    - Written in C: easily portable

Desktop PCs (1975-1990's)

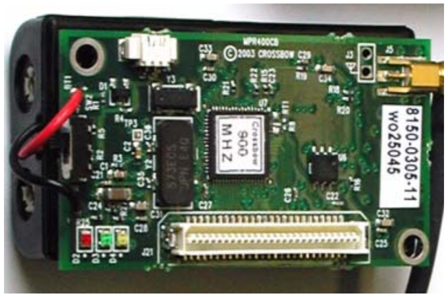
- Hardware technology
  - microprocessors
- The Experience
  - MAC (1980's): Steve Jobs discovers the Graphical User Interface Pioneered at Xerox-PARC
  - Microsoft catches up (1985-1995)
  - MINIX is born (1987) followed by Linux (1992)
- Automate office work

Internet PCs and Mobile Internet (1990’s - now)

- PCs connected using Internet
- Mobile phones connect you to Internet all the time
  - iOS, Android, …

Different modalities (1990's-now)

- **Parallel**: Multiple processors, one machine
- **Distributed**: Multiple networked processors
- **Real-time**: Strict or loose deadlines
- **Sensor networks**: **Many** tiny computers



Moral of the Story

- **The only constant: Change**
  - In 60 years, almost every computer component now 9 orders of magnitude faster, larger, cheaper
  - Example:

	1983	**1999**
MIPS	0.5	500
cost/MIP	$100,000	$500
memory	1 MB	1 GB
network	10 Mbit/s	1 GB/s
disk	1 GB	1 Tbyte

Moral of the Story, II

- No counterpart in any other sphere of human existence!
  - Transportation:
    - 200 years to go from horseback (10 mph) to Concorde (1000 mph) = 2 orders of magnitude
  - Communication is closest:
    - 100 years to go from Pony Express (10 mph) to nearly speed of light (600 million mph) = 7 orders of magnitude
- And operating systems must adapt…

Course outline

- Introduction
- Process management
  - Multiprogramming, process/thread, CPU scheduling, synchronization, deadlock
- File system
- Memory management
  - segmentation, paging, swapping
- I/O system
- Advanced topics
  - Protection, Security, etc.