Operating Systems
[0. Course Introduction]

Chung-Wei Lin
cwlin@csie.ntu.edu.tw
CSIE Department
National Taiwan University

Introduction to Myself

- B.S. Student
  - 2001.09--2005.06, CSIE Department, NTU
- M.S. Student
  - 2005.09--2007.06, GIEE (EDA Group), NTU
- Ph.D. Student
  - 2009.08--2015.08, EECS Department, UC Berkeley
- Researcher
  - 2015.09--2018.07, Systems and Software Division, Toyota InfoTechnology Center (Mountain View, CA)
- Assistant Professor
  - 2018.08--2021.07, CSIE Department, NTU
- Associate Professor
  - 2021.08--, CSIE Department, NTU

And You Are?

- ❑ Department?
- ❑ Year?
- ❑ Enrolled or auditing?
- ❑ Waiting list?
- ❑ Career goal?

Reasons of Teaching This Course

- What is a system?
  - "A system is a group of interacting or interrelated entities that form a unified whole" [Wikipedia]
  - The term "system" comes from the Latin word systēma, in turn from Greek σύστημα systēma: "whole concept made of several parts or members, system", literary "composition". [Wikipedia]
- Some "systems"
  - Operating systems (OSs)
  - Digital systems
  - Computer systems
  - Embedded systems
  - Cyber-physical systems

Reasons of Taking This Course (1/3)

- Get required units to graduate
- Transfer to CSIE?
- Prepare for graduate entrance exams?
- Learn fundamental knowledge of "operating systems"
- Increase your job-market value
  - You should be better than a pure software programmer
  - Software is running on hardware, and operating systems are between them

Reasons of Taking This Course (2/3)

- What you should know from System Programming Design
  - How to write programs using C (or C++) on a Linux environment
  - Using different system functions/libraries to implement a same function can have very different performance
  - Operating systems hide lots of details of executions and implementations
- What you may still wonder
  - How do those system functions/libraries are implemented?
  - Why do those system functions/libraries are implemented in this way?
  - What have operating systems done?
  - Can we do it differently?

Reasons of Taking This Course (3/3)

- A good program needs to run on an OS or hardware
  - Difficult to steal the root?
    - The Memory Sinkhole - Unleashing An X86 Design Flaw Allowing Universal Privilege Escalation
    - https://www.youtube.com/watch?v=lR0nh-TdpVg

Operating Systems

What is an OS?

- An OS is system software that manages computer hardware, software resources, and provides common services for computer programs [Wikipedia]

Examples

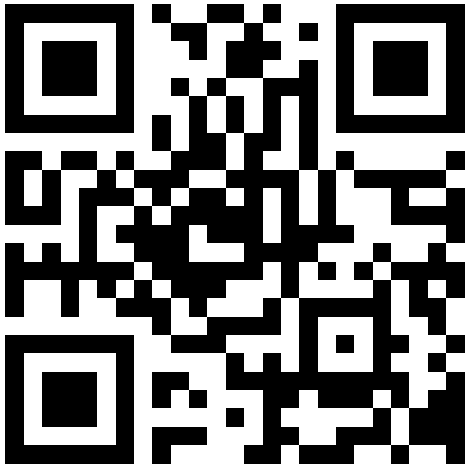
- Unix, BSD, macOS, Linux, Windows
- iOS, Android

OSs are complex

- One complex and complete OS can take million lines of code
  - MacOS 10.4@2005: 86 millions of source lines of code (SLOC)
  - Debian 7.0@2012: 419 millions of SLOC
  - Linux Kernel 4.2@2015: 20.2 millions of SLOC
- Note
  - Linux Kernel 0.01@1991: 0.010239 millions of SLOC

Enrollment and Auditing

- Submit your request at https://0rz.tw/flGmd
  - Due at 11:59pm, February 16 (Wednesday)



- Switching the enrollment between sections is not allowed
  - You are free to directly attend another section
- Auditing in-person lectures is not allowed
  - NTU students can submit requests (above) and access videos

Websites

- Course material, announcement, and machine problems (MPs)
  - NTU COOL: https://cool.ntu.edu.tw/courses/11767
  - **You are mandatory to check the announcement there**
- Report submission and grading
  - Gradescope: the link will be provided with a following MP

Textbook and References

❑ Textbook

- ➢ Avi Silberschatz, Peter Baer Galvin, and Greg Gagne, "Operating System Concepts," Tenth Edition, John Wiley & Sons
- ➢ Source codes: https://github.com/greggagne/osc10e

❑ References

- ➢ Russ Cox, Frans Kaashoek, and Robert Morris, "xv6: a simple, Unix-like teaching operating system," 2020
  - https://pdos.csail.mit.edu/6.828/2020/xv6/book-riscv-rev1.pdf
- ➢ W. Richard Stevens and Stephen A. Rago, "Advanced Programming in the UNIX Environment," Third Edition, Addison-Wesley
- ➢ Bruce Molay, "Understanding UNIX/LINUX Programming: A Guide to Theory and Practice," Prentice Hall

Lecture Schedule (Tentative)

Week	Content	Textbook (OSC)	Reference (xv6)	MP Assignment	MP Due
1 (Feb 15)	Course Introduction		Ch0	xv6 Setup	
2 (Feb 22)	Overview	Ch1, Ch2			
3 (Mar 1)	Processes, Threads, and Concurrency	Ch3	Ch1		MP0 Due
4 (Mar 8)		Ch4		Process/Thread	
5 (Mar 15)	Memory Management	Ch9	Ch2		
6 (Mar 22)		Ch10		Memory	MP1 Due
7 (Mar 29)	I/O Systems	Ch12			
8 (Apr 5)	Spring Break				MP2 Due
9 (Apr 12)	Midterm				
10 (Apr 19)	CPU Scheduling	Ch5	Ch7		
11 (Apr 26)				Scheduling	
12 (May 3)	Mass-Storage and File System	Ch11, Ch13	Ch8		
13 (May 10)		Ch14, Ch15		File Systems	MP3 Due
14 (May 17)	Synchronization	Ch6	Ch6		
15 (May 24)		Ch7			MP4 Due
16 (May 31)	Final Exam				
17 (Jun 7)	TBD				
18 (Jun 14)					

Grading

- Midterm: 28% [Week 9]
- Final Exam: 28% [Week 16]
- 5 MPs: 44%
  - 4% + 10% + 10% + 10% + 10%
- Bonus
  - Submit a short self-review report with the MP4 report
  - Provide "evidence" (e.g., a post on NTU COOL) how you helped others

- Both sections will have the same exams and the same MPs
- Final grades may be adjusted at different sections
  - The original averages were very close last year
  - Expected grades of attending another section are the same

Machine Problems (MPs)

- TAs: 林祥瑞, 劉昕佑, 曾奕青, 林家佑, 周良冠, 賈本晧
  - If you have a question, you should ask it on NTU COOL
  - Email: ntuos@googlegroups.com
    - This email is only for you to ask questions that are private
    - We will not answer questions that should be asked on NTU COOL
- MP0: set up xv6 (announced today)
- MP1--4: thread, shared memory, scheduling, file systems

- Bonus and penalty
  - Early bird bonus may be applied
  - Late penalty will be applied after submission deadline
    - Example: 20%-off per day

Plagiarism

- There is NO tolerance for plagiarism
- You can discuss with your classmates, but you must write the codes on your own
- It is your responsibility to protect your own codes
  - Do not leave your codes on tables, screen, or unlocked directory
  - Change the permission of your home directory on workstations
  - Create private repositories on GitHub
- We will cross check your codes with others'

My Own Learning Experience

- Concepts
- Mechanisms and behind reasons
  - Textbook
  - No pain, no gain
- Engineering course?
- Evolution

Q&A