Operating Systems

Operating Systems

Wenbo Shen

whoami

- Wenbo Shen (申文博)
  - A Zhejiang University 100 Young Professor
  - A system security researcher
  - A kernel programmer
- R&D highlights
  - Tech lead of Samsung Knox Kernel, Silicon Valley (4 years)
    - Design and implement features protecting 100+ million flagship devices
  - Control flow protection: first in mobile industry, shipped in 2016
    - Google Pixel catches up partially by end of 2018
  - Publications in all top4: IEEE S&P, ACM CCS, USENIX Sec, NDSS
    - TZ based RKP(CCS 14), SKEE (NDSS 16), XOM (S&P 17), Pex(USENIX Sec19)
- Education
  - PhD from North Carolina State University, USA, 2015
  - BE from Harbin Institute of Technology (哈工大), Harbin, 2010

Course Info

- Instructor - Wenbo Shen
    - https://wenboshen.org/
    - shenwenbo@zju.edu.cn

- Class hour: Mo, Th 14:05-15:40; Sa 15:55-17:30

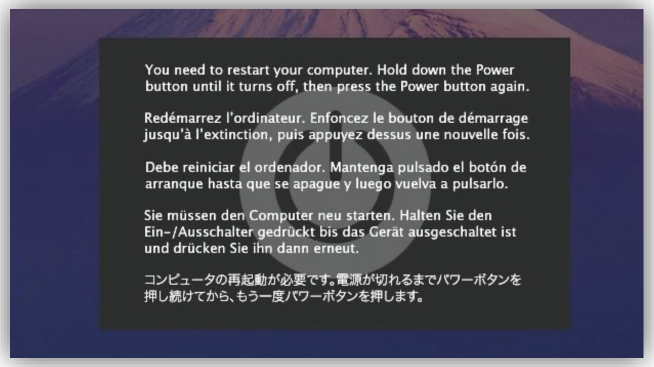
- Office hour: Sa 15:55-17:30

- TA:
    - Jinmeng Zhou (1st year PhD student)
    - Jiadong Sun (1st year Master student)

- Class website: TBD

Why are we studying OS?

- OS is highly complicated software running on most machines
  - Windows: 50M lines of source code
  - Linux: 25M lines of source code
- It contains many important system concepts
  - complexity hiding, performance tuning, resource allocation…
- Studying OS internals makes you a more **capable** programmer
  - know how it works, and how it works better


You need to restart your computer. Hold down the Power button until it turns off, then press the Power button again.

Redémarrez l'ordinateur. Enfoncez le bouton de démarrage jusqu'à l'extinction, puis appuyez dessus une nouvelle fois.

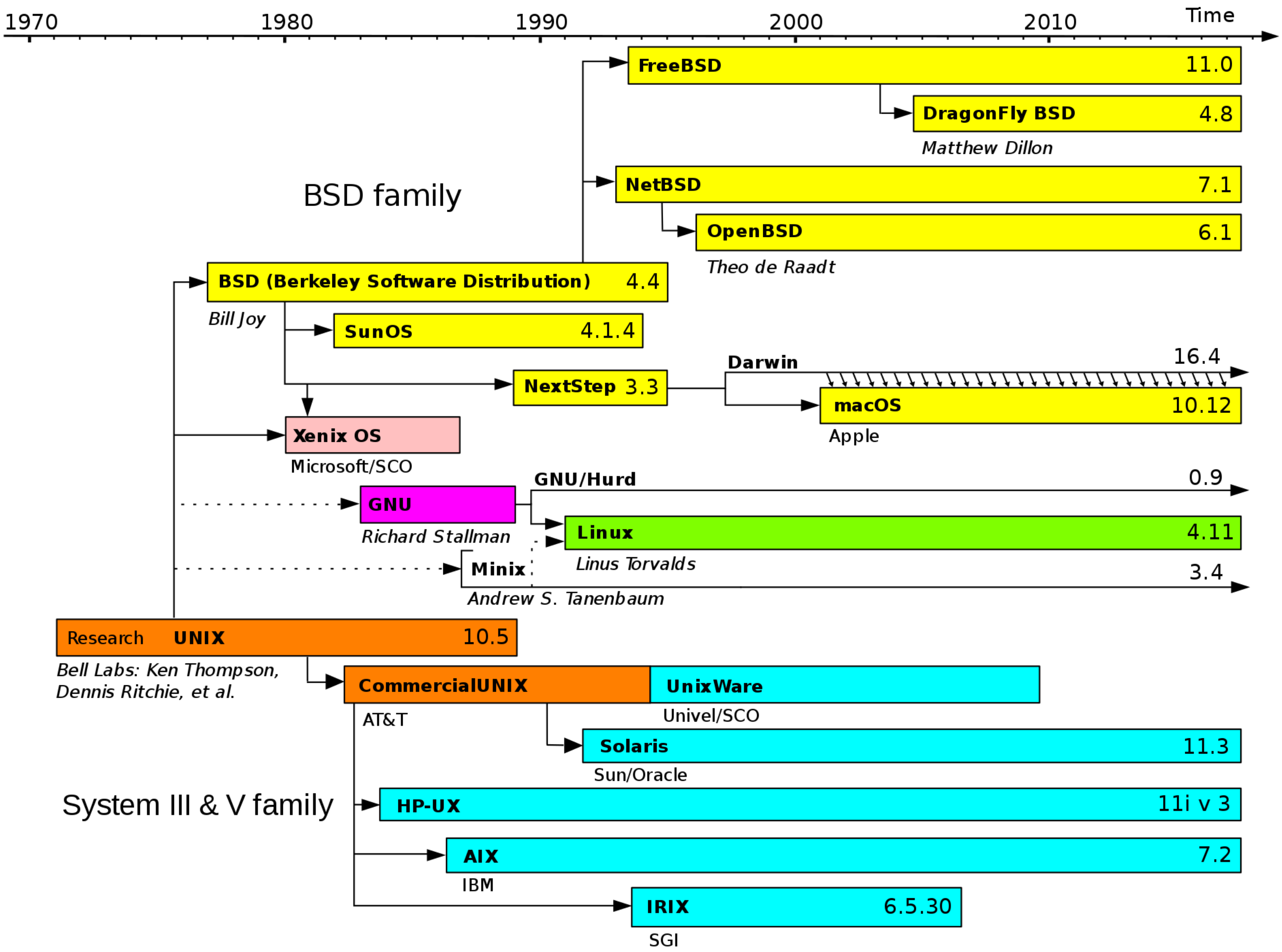
Debe reiniciar el ordenador. Mantenga pulsado el botón de arranque hasta que se apague y luego vuelva a pulsarlo.

Sie müssen den Computer neu starten. Halten Sie den Ein-/Ausschalter gedrückt bis das Gerät ausgeschaltet ist und drücken Sie ihn dann erneut.

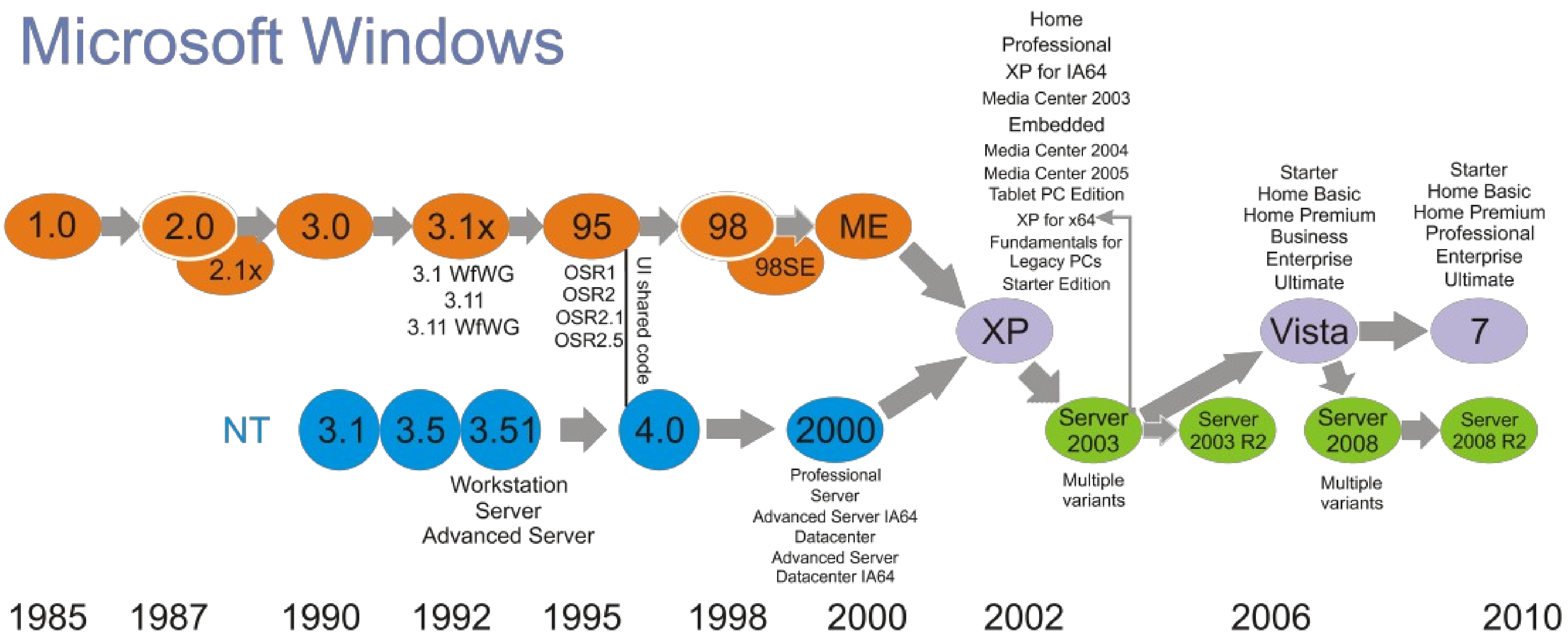
コンピュータの再起動が必要です。電源が切れるまでパワーボタンを押し続けてから、もう一度パワーボタンを押します。

Popular Operating Systems

UNIX Family Tree



Windows Family Tree



Why are we studying OS?

- After all, you probably will not develop an OS

Why are we studying OS?

- ~~After all, you probably will not develop an OS~~
  - How about first OS of China, I mean the REAL OS

Why are we studying OS?

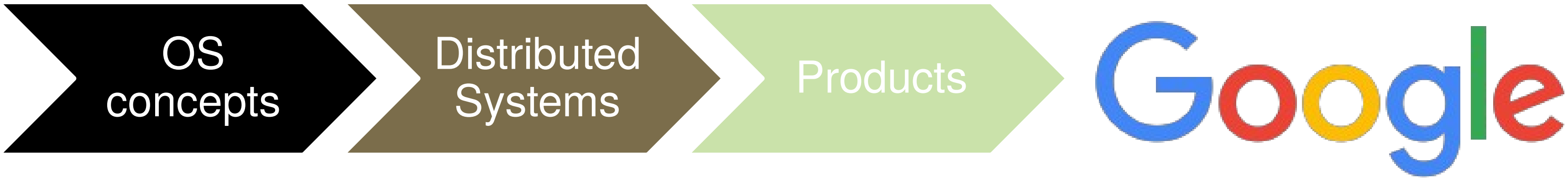
- ~~After all, you probably will not develop an OS~~
  - How about first OS of China, I mean the REAL OS
- OS concepts benefit whole life
  - OS concepts are re-usable when implementing other software
  - Lessons learned from OS study can be applied to complex software systems, such as map-reduce, DNS
- Foundation of ALL software
  - Better user-space software, including apps
    - Invoke proper kernel API
    - What can and cannot be done
    - Inconsistent with kernel design
  - Better performance
    - Caching

Why are we studying OS?

- Non-textbook answers
  - For know-everything-feeling (demo)
  - For Hacking
    - The more you know OS, the better hacker you are
    - Because the thing you are trying to hack into, probably is running an OS

Why are we studying OS?

- Non-textbook answers
  - For Profit
    - Interview = coding + system design
    - Great System == Great Product == Great Company


OS concepts

Distributed Systems

Products

Google MapReduce

Google File System

Google Cloud

Fuchsia by Google

ANDROID

Wear OS by Google

Google Chrome OS

Learning Objectives

- Understand operating system concepts
  - process management, CPU scheduling, synchronization, file systems…
  - multi-threading and synchronization, system call, kernel modules…

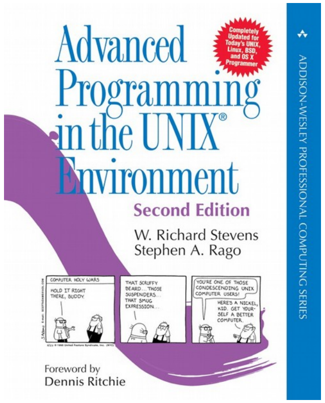
- Get an overall understanding of how the real-world operating systems work

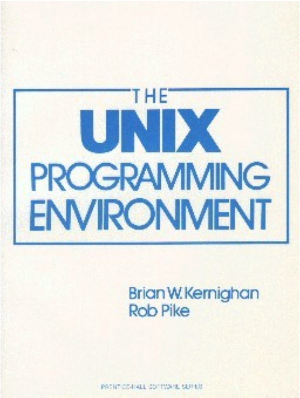
Learning Objectives

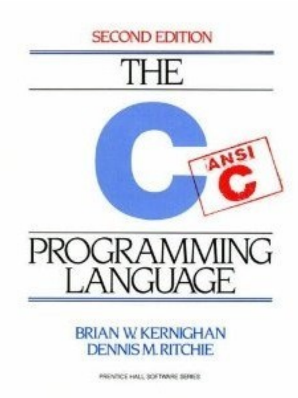
- Understand operating system concepts
  - process management, CPU scheduling, synchronization, file systems…
  - multi-threading and synchronization, system call, kernel modules…
- Get an overall deep understanding of how the real-world operating systems work
  - You can never truly understand a concept unless you implemented (CODE) it

Prerequisites

- Data Structures
- Programming skills:
  - proficiency in UNIX(Linux) programming and debugging
  - proficiency in the C programming language
- Or you are willing to learn


Advanced Programming in the UNIX® Environment
Second Edition
W. Richard Stevens
Stephen A. Rago
Foreword by Dennis Ritchie
Completely Updated for Today's UNIX, Linux, BSD, and OS X Programmer
ADDISON-WESLEY PROFESSIONAL COMPUTING SERIES

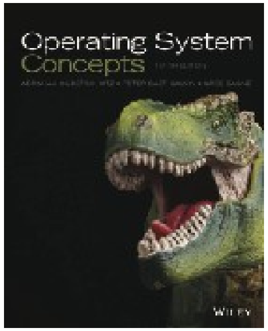




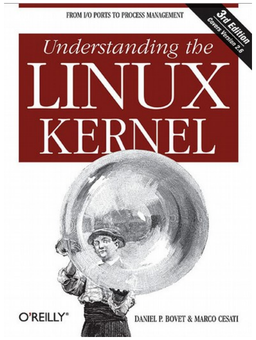
- Or you are willing to learn

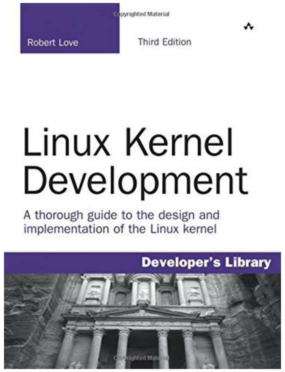
Course Material

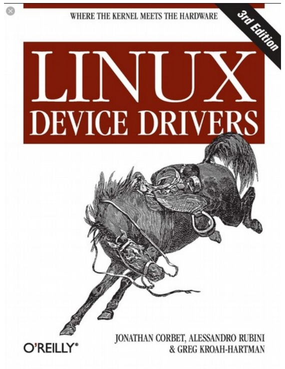
- Lecture notes (posted at the class website)
- Textbook: Operating System Concepts



- Very useful it you do Linux kernel programming:


FROM I/O PORTS TO PROCESS MANAGEMENT
3rd Edition
Understanding the
LINUX
KERNEL
O'REILLY
DANIEL P. BOVET & MARCO CESATI


Robert Love
Third Edition
Linux Kernel
Development
A thorough guide to the design and implementation of the Linux kernel
Developer's Library


WHERE THE KERNEL MEETS THE HARDWARE
3rd Edition
LINUX
DEVICE DRIVERS
O'REILLY
JONATHAN CORBET, ALESSANDRO RUBINI
& GREG KROAH-HARTMAN

Grading

Regular Track

- Final Exam – 50%
- Homework – 10%
- Survey/Presentation – 12%
- Class Quiz – 7%
- Project – 21%

Advanced Track

- Final Exam – 50 points
- Homework – 10 points
- Project – 40+ bonus points

Homework and Projects

- Three projects + one bonus project (Regular Track)
- Experimental project – Porting Linux0.11/0.12 to ARM64 and RISC-V (Advanced Track)
- Project (Regular) and homework assignments are **individual** efforts
- Submission **MUST** be typed, no hand-written submission
- Late submissions are accepted **after the deadline**
  - a **10%** penalty will be applied for **each day** of late submission
- Disputes of grade MUST be resolved within **one week** of receiving it

Exams

- No midterm exam
- One final exam, close book
- Final exam is comprehensive/cumulative

Your Responsibilities

- Understand lecture & reading materials
- Ask for extra help (talk to me or TA), if needed
  - if the class is too hard or you do not have necessary backgrounds
- Uphold academic integrity
- Turn in your assignments on time
- Check class web page regularly

Dos and Don’ts

- Do share debugging experiences, knowledge of tools
- Do acknowledge help from others
- Do acknowledge sources of information from books and web pages

- Don’t cheat or help others cheat
- Don’t share code from others
  - e.g., changing variable names or indentation
- Don’t post code to the discussion board

Cheating policy

- Cheating is not allowed
- We run tools. If you cheat, you will probably get caught. If you get caught, you will get a negative score on the assignment and likely fail the whole course (not just for the parts you were caught cheating)
- I REFER ALL ACADEMIC DISHONESTY INCIDENTS TO THE OFFICE OF STUDENT CONDUCT, WITHOUT EXCEPTION
- If you don’t cheat and work hard, you will always do better than if you cheated
- Draw your own figures, use your own words and add the citation

This includes the research project!

All text and figures should be your own.

Take away

- OS is the foundation of ALL software
- OS concepts benefit the whole life
- OS concepts are hard to understand unless you code

- Don’t cheat
  - Give proper credits to others’ work

Questions

- Questions?