

Operating Systems



Operating Systems
Wenbo Shen

■ 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(ACS 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

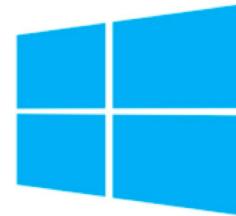
- Instructor - Wenbo Shen
 - <https://wenboshen.org/>
 - shenwenbo@zju.edu.cn
- Class hour: Wed, Fri 08:00-09:35
- Lab hour: Fri 09:50-11:25
- TA:
 - Ade Zhuang (2nd year Master student)
- Class website: TBD

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 hacker
 - know how it works, and how it works better



Popular Operating Systems



Windows 10



Mac OS X

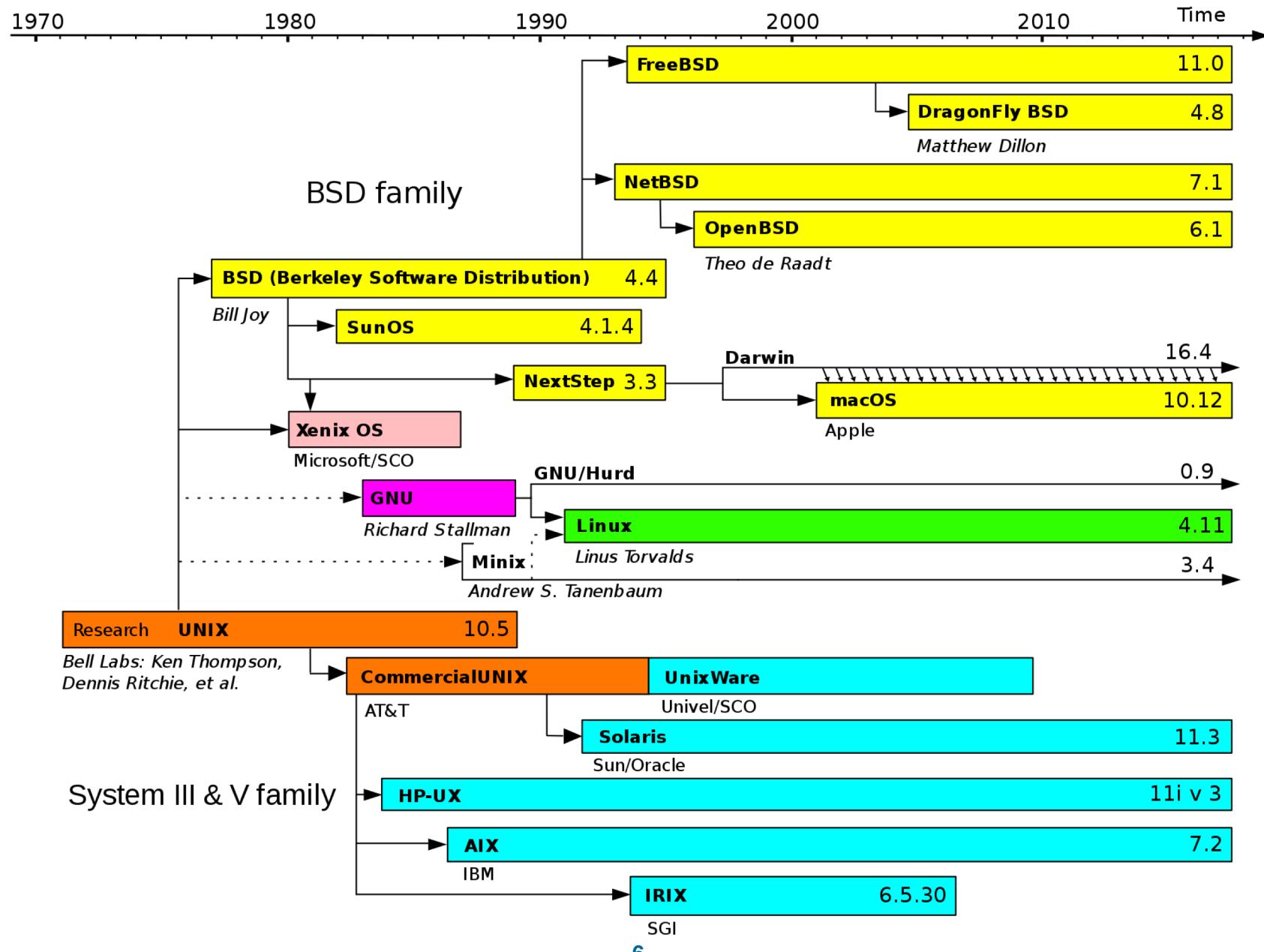


Windows Phone



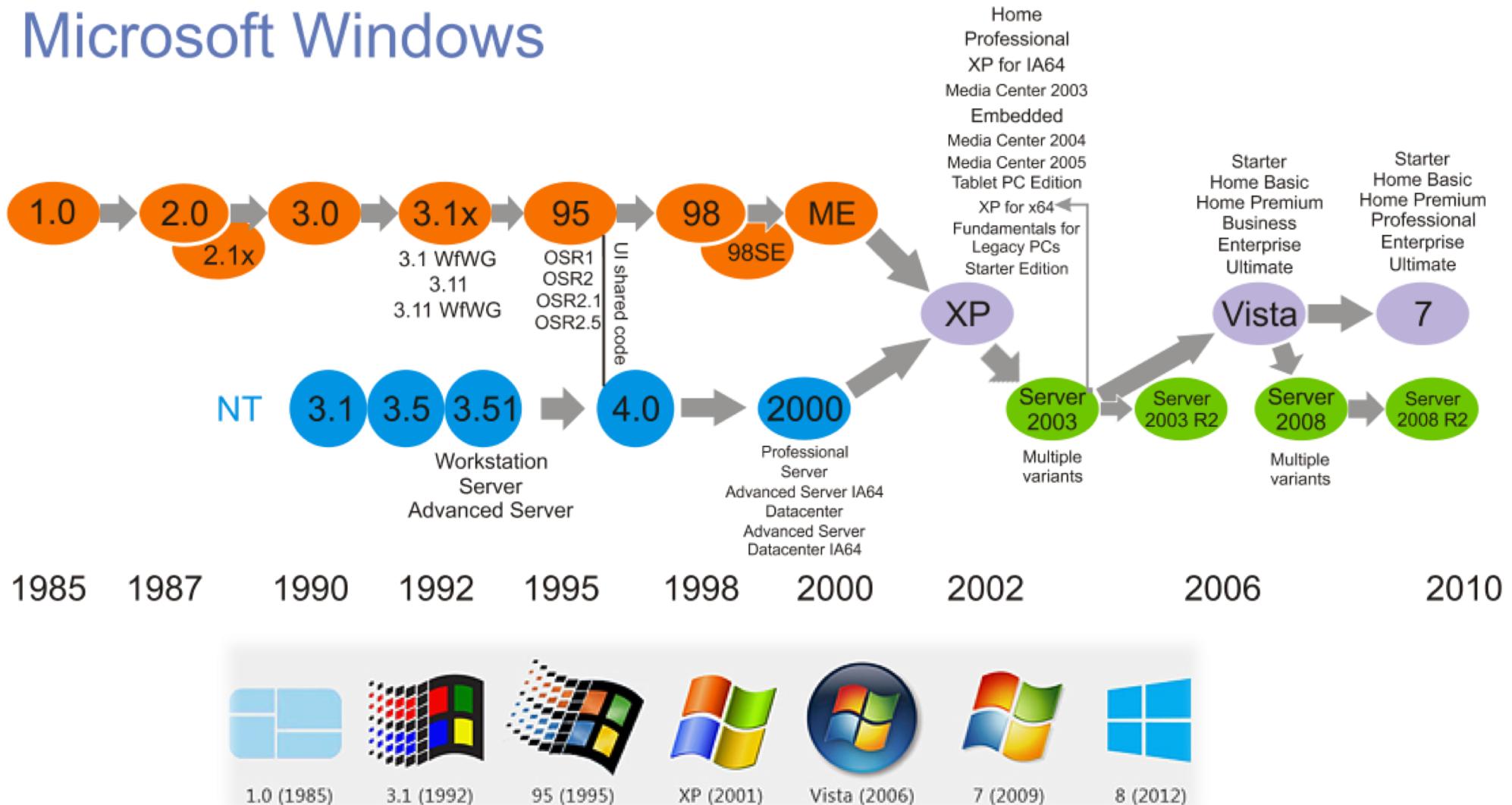
BlackBerry

UNIX Family Tree



Windows Family Tree

Microsoft Windows



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
 - Better performance
 - ▶ Caching
 - ▶ Scheduling
 - ▶ Memory management

Why are we studying OS?

■ Non-textbook answers

- For know-everything-feeling
- 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



Google MapReduce

Google File System

Google Cloud

Fuchsia
by Google

Android

Wear OS by Google

Google Chrome OS

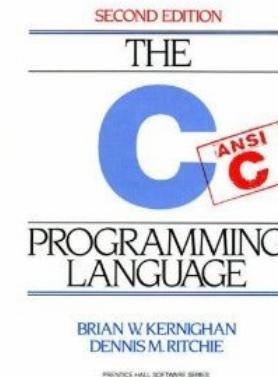
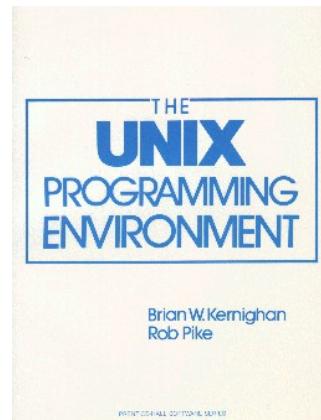
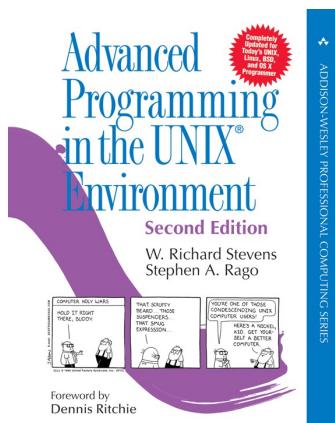
Learning Objectives

- Understand operating system concepts
 - process management, CPU scheduling, multi-threading, synchronization
 - memory management, physical memory, virtual memory
 - file systems...

- Get a deep understanding of how the real-world operating systems work
 - You can never truly understand a concept unless you **implemented (CODE)** it

Prerequisites

- Asembly, C, Data structures
- Programming skills:
 - proficiency in the C programming language
 - proficiency in UNIX(Linux) programming and debugging

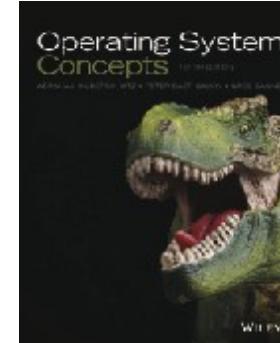


- Or you are willing to learn

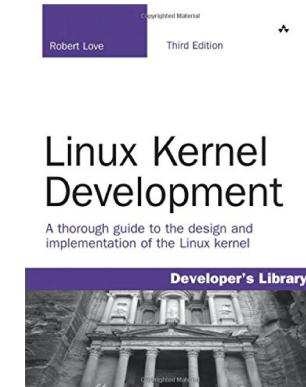
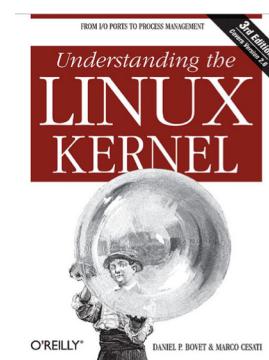
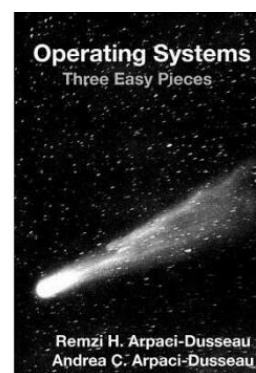
Course Material

- Lecture notes (posted at the class website)

- Textbook: Operating System Concepts



- Very useful if you do Linux kernel programming:

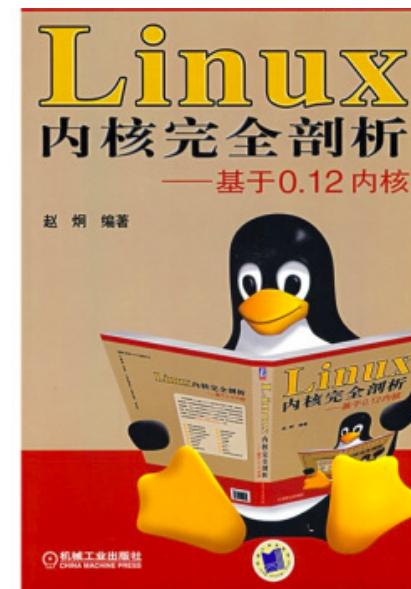
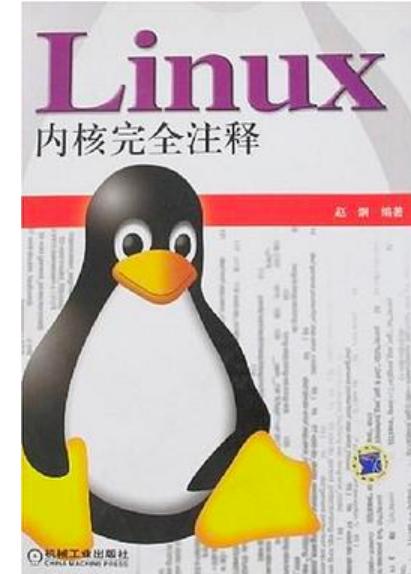


Grading

- Final Exam – 50 points
- Homework – 10 points
- Project – 40 points + bonus
 - You are in advanced class



- Implement Linux0.11 on RISC-V 64-bit
- One bonus project
- Details TBD



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

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?