

Videos

Overview

Videos and notes from class will be posted here.

Lecture 1: Introduction to the class, including the basics of what an OS does, and the major themes of the course: virtualization, concurrency, and persistence. Overview of how the class works. Finally, introduction to CPU virtualization, and start of discussion on mechanisms to support it. [Notes](#) - Video: [Part 1](#) - [Part 2](#) - [Part 3](#)

Discussion 1: Introduction to project p1a, some simple Unix utilities. [Video](#) - [Code](#)

Lecture 2: Recapitulation of mechanisms to support CPU virtualization, and completion of said topic. Introduction to CPU scheduling (policies). Introduction to homeworks for CPU virtualization chapters. [Notes](#) - Video: [Part 1](#) - [Part 2](#) - [Part 3](#)

Discussion 2: Introduction to project p1b, adding a system call to xv6. [Video](#)

Lecture 3: CPU scheduling policy (continued). Focus on multi-level feedback queue (Part 1). Introduction to virtual memory, including simplest mechanism called dynamic relocation (Part 2). Wrap up dynamic relocation, talk about segmentations (Part 3). [Notes \(1,2\)](#) [Notes \(3\)](#) - Video: [Part 1](#) [Part 2](#) [Part 3](#)

Discussion 3: Introduction to project p2a, the Unix shell. [Video](#)

Lecture 4: Review of dynamic relocation and segmentation, and problems. Intro to paging (Part 1). More on paging mechanisms, and intrinsic problems (Part 2). Introduction to TLBs (Part 3). [Notes](#) - Video: [Part 1](#) [Part 2](#) [Part 3](#)

Lecture 5: Review of paging and its problems. How to handle the fact that linear page tables are too big? Hint: multi-level page tables (Part 1). How to handle the fact that programs might use more memory than is physically available? Hint: swap to disk. (Part 2). [Notes](#) - Video: [Part 1](#) [Part 2](#)

Discussion 5: Intro to xv6 scheduler. [Video](#)

Lecture 6: Intro to concurrency (Part 1). Understanding a simple data race, the lack of mutual exclusion, and indeterminate outcomes (Part 2). How to build a simple spin lock using atomic exchange, how to add locks to a list or hash table (Part 3). Notes: [Part 1](#) [Part 2](#) - Video: [Part 1](#) [Part 2](#) [Part 3](#)

Discussion 6: C fun and games. [Video](#) - [Code](#)

Lecture 7: More concurrency. Review of spin locks and their problems. Ticket locks, yield(), and park()/unpark() (Part 1). Wrap up of locks (for now). Introduction to condition variables, and the fork/join problem (Part 2). Continued fork/join, intro to bounded buffer (producer/consumer) problem

(Part 3). Handouts: [Fork/Join Producer/Consumer](#) - Notes: [Parts 1,2](#) [Part 3](#) - Video: [Part 1](#) [Part 2](#) [Part 3](#)

Discussion 7: Intro to p3. [Video](#)

Lecture 8: Remembering producer/consumer (Part 1). Ending producer/consumer, intro to semaphores (Part 2). [Notes](#). Handouts: [Semaphores](#). Video: [Part 1](#) [Part 2](#) [Part 3](#) ([Discussion](#))

Lecture 9: Bugs and deadlock (Part 1). Review (Part 2). [Notes](#). [Part 1](#) [Part 2](#)

Review: [Notes](#). [Video](#)

Lecture 10: Introduction to Persistence. The Hard Disk Drive. Disk Scheduling. [Notes](#). - Video: [Part1](#) [Part2](#) [Part 3](#)

Discussion 10: xv6 and Thread Creation. [Video](#)

Lecture 11: RAID. Levels 0, 1, 4, and 5. File System API and Abstractions. Notes: [1](#) [2](#) - Video: [Part 1](#) [Part 2](#) [Part 3](#) and [Part 4](#)

Discuss 11: Brief intro to MapReduce. [Notes](#) (same as above) - Video: [Part 1](#)

Lecture 12: Intro to on-disk file system structures (Part 1). Access methods (Part 2). Caching + Stuff (Part 3). [Notes](#) - Video: [Part 1](#) [Part 2](#) [Part 3](#)

Discuss 12: Q/A on MapReduce. Video: [Part 1](#)

Lecture 13: Intro to crash consistency (Part 1). Crash consistency and journaling (Part 2). Fast File System (Part 3). [Notes](#) - Video: [Part 1](#) [Part 2](#) [Part 3](#)

Lecture 14: LFS intro (Part 1). More LFS, including garbage collection (Part 2). Flash-based SSDs (Part 3). So-called words of wisdom (Part 4). [Notes](#) - Video: [Part 1](#) [Part 2](#) [Part 3](#) [Part 4](#)

Review: [Notes](#) - Video: [Part 1](#) [Part 2](#)

.