ILLINOIS TECH

College of Computing

CS 450 Operating Systems Course Overview

Yue Duan

CS 450 Operating Systems

- Instructor:
  - Yue Duan, Assistant Professor
    - https://yueduan.github.io/
    - yduan12@iit.edu
  - TAs:
    - Yihua Xu
      - yxu131@hawk.iit.edu
    - William Bodell
      - wbodell@hawk.iit.edu

CS 450 Operating Systems

- Lecture Hours
  - Tu/Th 8:35am - 9:50am
- Lecture location
  - SB 104 + Blackboard Collaborate Ultra

Classes on the week of Jan 24 may still be online via Blackboard

CS 450 Operating Systems Office Hour

- Yue Duan
  - Office: SB 209C, Wednesday 3 - 5pm
  - https://iit-edu.zoom.us/j/2914428434
- Yihua Xu
  - SB 013b, Tu/Wed 1:00pm - 2:30pm
  - https://iit-edu.zoom.us/j/89004458281?pwd=aWpxd3VrN2VSQkZ2T29majF3UVhHdz09
- William Bodell
  - SB 013b, Thursday 1:00pm - 4:00pm
  - https://iit-edu.zoom.us/j/86774394546?pwd=cktQUFZpei9BaUZNMThINXFVQnI3QT09

CS 450 Operating Systems

- Communication:
  - We will use piazza as our major communication mechanism
  - https://piazza.com/iit/spring2022/cs450
  - Please sign up if you haven’t
- Textbook:
  - (Free!) Operating Systems: Three Easy Pieces, Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau
  - A good resource for lab assignments:
    - XV6 Documentation

CS 450 Operating Systems

- Course webpage
  - https://yueduan.github.io/cs450.html
  - syllabus, schedules, slides posted after each class
- Course Components:
  - Lectures
  - Exams
  - Lab Assignments
  - Homeworks

CS 450 Operating Systems

- Prerequisites
  - System programming skills
  - Some UNIX environment familiarity
  - Knowledge of data structures, algorithms, etc
- Course topics will cover but not limited to:
  - Process Scheduling/Synchronization
  - Memory Management
  - File System
  - Computer Security

CS 450 Operating Systems

- Grading
  - Lab assignments: 40%
    - four lab assignments
    - implement new features on top of XV6
  - Homework: 15%
    - three homework assignments
  - Midterm exam: 20% (Tentative: 2/24)
  - Final exam: 25% (TBD)
- Goal is give everyone **A**

CS 450 Operating Systems

- Lab assignments:
  - Lab1: Environment setup
  - Lab2: Fun with system calls
  - Lab3: Process management
  - Lab4: Memory management
- Late policy:
  - within one day after the deadline ===> lose half points.
  - Submissions one day after the deadline will **NOT** be accepted
  - unless you get permission from the instructor

Tentative course schedule

- **1.11** Intro and administrivia
- **1.13** OS and xv6 overview
- **1.18 - 1.20** Process, thread and scheduling
- **1.25 - 2.10** Synchronization (**2.3** Lab1 due)
- **2.15 - 3.8** Memory management (**2.24** Mid-term, **3.1** Lab 2 due)
- **3.22 - 3.31** I/O (**3.29** Lab3 due)
- **4.5 - 4.14** File Systems
- **4.19 - 4.28** Network and security (**4.28** Lab4 due)
- **TBD** Final

CS 450 Operating Systems

Acknowledgement

Some materials from Prof. Heng Yin from UCR, Prof. Fred Schneider and Prof. Robbert Van Renesse from Cornell, and Prof. Kyle Hale from IIT

What an OS does

- OS is an intermediary between programs and hardware.
- OS creates an environment to execute programs in a convenient and efficient manner:
    - allocates resources (CPU and storage)
    - controls programs
        - cooperation (sharing and synchronization)
        - isolation (protection and resource management)

What an OS does

What an OS does

- Ways to view OS
  - **Services** it provides to programs
  - **Components** implementing those services
    - internal design and implementation
    - Real hardware is ugly
  - interactions

Why study OS

- Learn solutions to problems arising in all systems:
  - Resource sharing (scheduling)
  - Cooperation (concurrent programming: communication, synchronization)
  - System structure (abstractions, interfaces)

System vs Programs

- Measure of success:
  - OS concerned with extensibility, security, reliability, …
- External interface:
  - OS more complicated and subject to change. e.g., I/O devices.
- Internal structure:
  - OS must pick boundaries between functions.
    - Heap with garbage collection vs stack allocation
    - RISC vs CISC

System vs Programs

- Structuring techniques:
  - OS employs - modules, layers, client-server, event-handler, transaction
- OS must bridge mismatched performance characteristics.
  - Registers vs RAM vs Disk
  - Desktop processor vs Cloud

Manage Complexity

- Modularity:
  - Good modularity minimizes connections between components.
- Abstraction:
  - Separate interface from internals
  - separate specification from implementation
- Hierarchy:
  - Constrain interactions so easier to understand.
  - L levels with N components per level
  - $(N \times L)^2$ vs $L \times N^2$ interactions

OS Roles

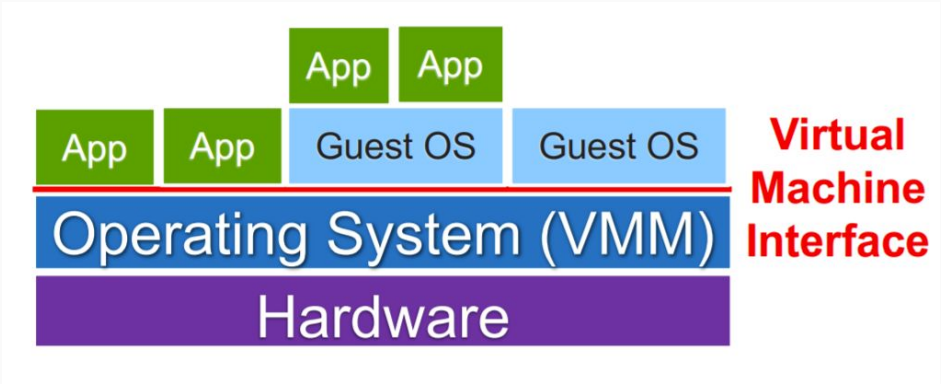
- Referee
  - Manages shared resources: CPU, memory, disks, networks, displays, cameras, etc.
- Illusionist
  - Look! Infinite memory! Your own private processor!
- Glue
  - Offers set of common services (e.g., UI routines) • Separates apps from I/O devices

OS as Referee

- Resource allocation
  - Multiple concurrent tasks, how does OS decide who gets how much?
- Isolation
  - A faulty app should not disrupt other apps or OS
  - OS must export less than full power of underlying hardware
- Communication/Coordination
  - Apps need to coordinate and share state

OS as Illusionist

- Virtualization: Resources seem present but aren't.
  - processor, memory, screen space, disk, network
  - the entire computer:
    - ease of debugging, portability, isolation



OS as Glue

- Simplify app design and facilitate sharing due to:
  - send/receive of byte streams
  - read/write files
  - pass messages
  - share memory
  - UI
- Decouples HW and app development

THANK YOU!