CS111
Operating Systems Principles

Introduction

Mendel Rosenblum

http://cs111.stanford.edu

What is an Operating System?

- “Operating system” is a hard term to define
- The discipline arose historically from a set of problems
- It’s easiest to introduce OS’es by discussing their history

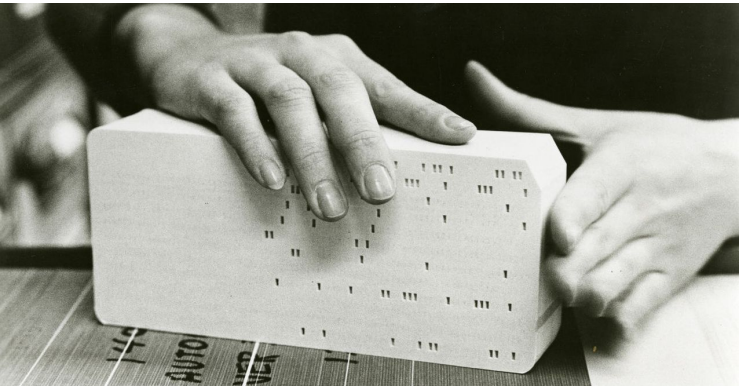
Learn OS Through History

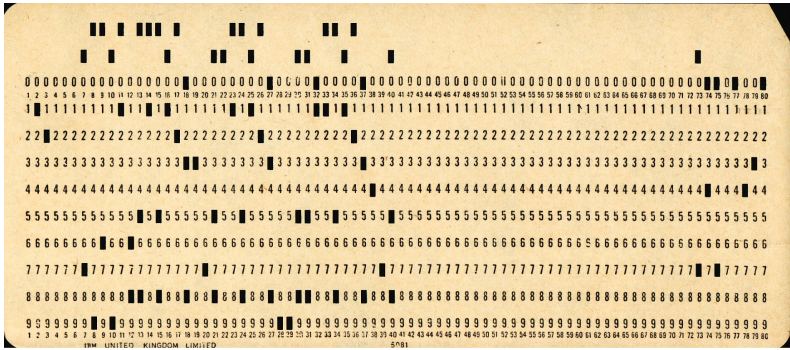
- OS evolved to solve real problems
- We’ll trace how features emerged
- Then extract principles

1940s Operating Systems

- Started with first computers, 1940s.
- One user at a time, working directly at console
- First “operating systems”: shared code for things like reading and writing devices (e.g. input/output libraries)
    - Reasons: convenience, efficiency

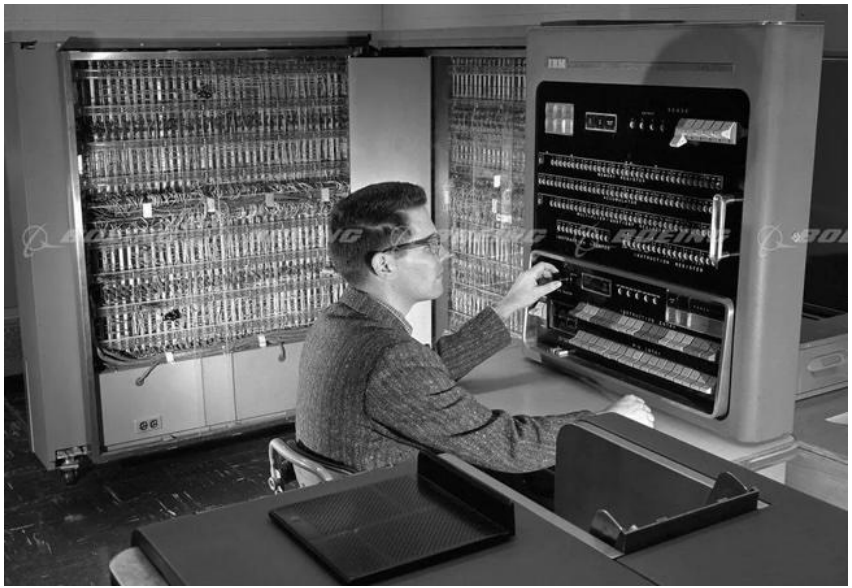
https://www.ibm.com/history/punched-card





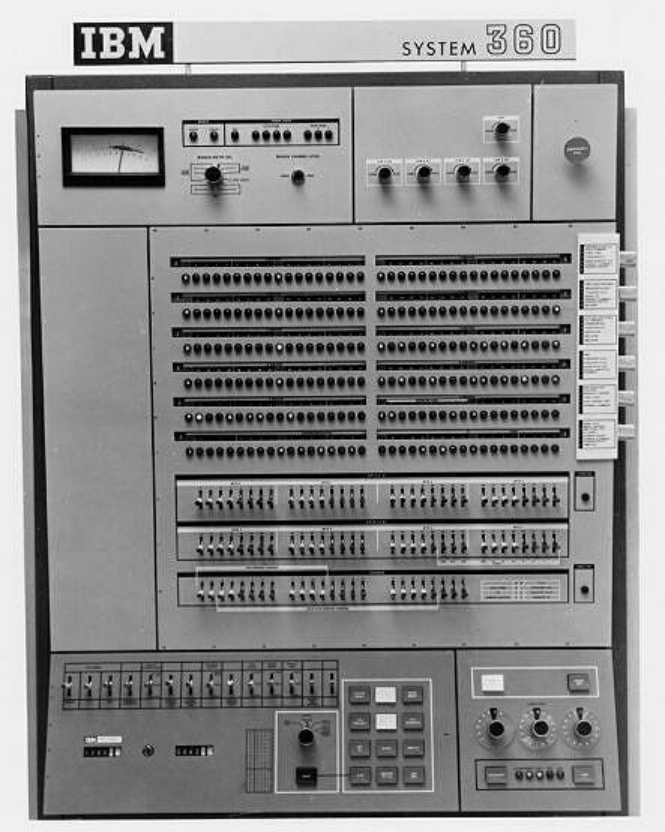
https://commons.wikimedia.org/wiki/File:Used_Punchcard_(5151286161).jpg

1952 - IBM Model 701 - First commercial computer



Vacuum Tubes, OS was shared cards

IBM System 360 Console - Debug interface

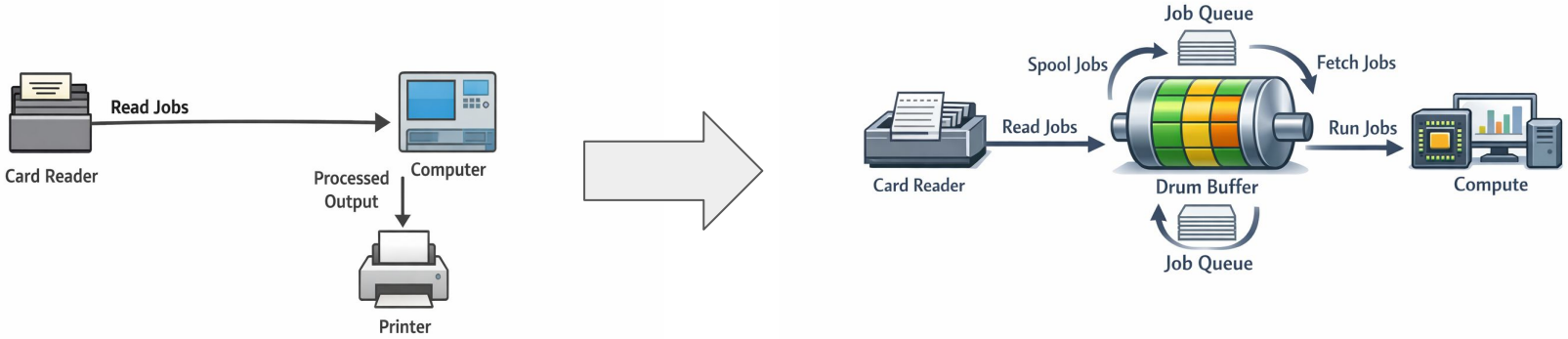


Phase 1: Hardware Expensive, Humans Cheap

- Goal: maximize machine utilization
  - Get user out of the loop
- Simple **batch monitor**:
  - User submits deck of punched cards describing a series of operations (job)
  - Output is given back as a print out
    - Program output
    - Print out of memory contents on errors (core dump)
- OS = program to load and run user programs and take memory dumps after crashes
  - Makes better use of hardware, but more difficult to debug.

1960s Improving Efficiency (machine utilization)

- Overlap of input/output (I/O) and computation
  - Hardware: data channel & interrupts
- Buffering and interrupt handling in OS
  - OS: Handling many things happening at the same time



Simple Batch Monitor Issues

- Still one job at a time → utilization often bad
  - Couldn't run something else while job was waiting for I/O
- No protection
  - Misbehaving job could damage batch monitor
- Short jobs wait for long ones
  - Ideally we could reorder jobs

1962 IBM 7094

- Transistor technology
  - Bigger main memory
- OS:
  - Memory relocation
  - Memory protection

Multitasking and Kernel

- Memory protection and relocation enable multitasking
- Several users share the system
- OS kernel emerges
  - Code that runs in privileged mode
  - Manages interactions, concurrency

Complexity + Emergence of Field

By mid-1960's operating systems had become large, complicated

Famous disasters (Multics, OS/360)

OS field emerges as important discipline

Software engineering developed out of problems with large operating systems

Phase 2: Hardware Cheap, Humans Expensive

- Make more efficient use of people time
- Get people back closer to hardware

Timesharing

- Interactive **timesharing**
  - File systems (convenience)
  - Issues of response time, thrashing (efficiency)

Bring user & computer together

- Personal computers: one per user
  - Started simple got complex

Networking

- 1993 World Wide Web
- Networking: sharing and communication between machines

Computers everywhere

- Personal devices: phones, TVs, light switches

Phase 3: Modern Systems - 2000s

- Very small (devices)
- Very large (datacenters, cloud)

What OSes Do Today

- Managing shared resources efficiently
- CPU: Concurrency: allow several tasks to share processors
- Memory: share memory among processes
- I/O devices: manage operations efficiently
- Files: share storage across users
- Networks: allow computers to work together
- Security: protect participants from each other

The OS Kernel

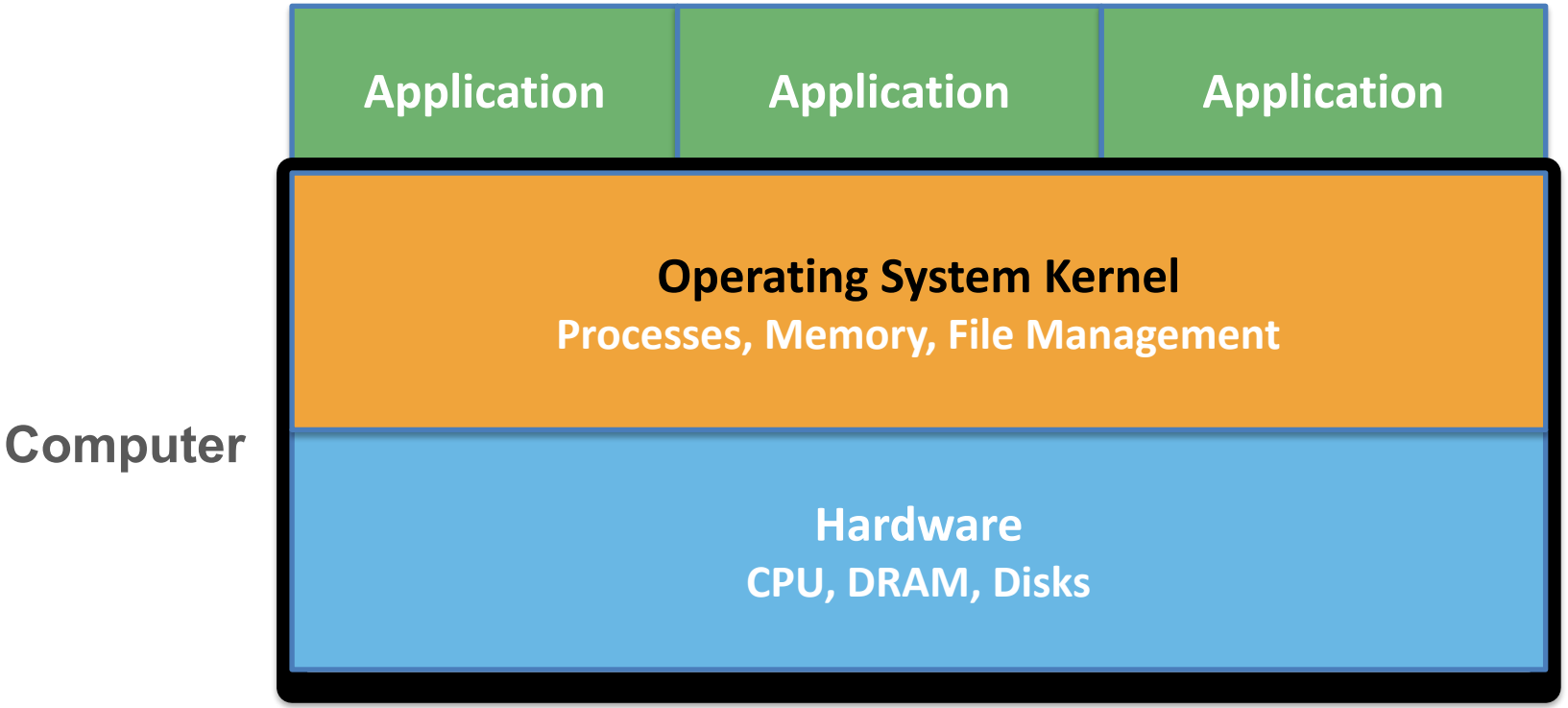
Code that implements these functions is called the **kernel**

Kernel code occupies reserved area of memory

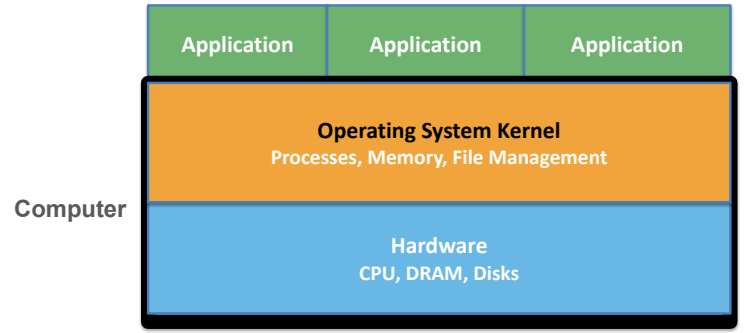
Has control over the entire system

User application code is restricted

Modern Computer System Structure



Why OS is Interesting



- Learn how things really work
  - OSes are magic
- Interesting concepts such as concurrency and virtual memory
- Integrates ideas from many areas of CS
- Philosophical issues:
  - Is fairness more important than overall happiness?
  - Does the past predict the future?

Announcements

Course Organization: Three parts

- Concurrency
  - Processes, synchronization, scheduling
  - 4 programming assignments
- Memory Management
  - Linkers, dynamic storage, virtual memory, paging
  - 2 programming assignments
- File Systems
  - Disk layout, directory structures, crash recovery
  - 2 programming assignments

Teaching Team

Mendel Rosenblum

Febie Lin (Head TA)

TBA

Patrick Jovel

Ricky Rios

James Varah

Alex Liu

Matthew Mattei

CS111 differences from CS106x and CS107

- Earlier courses (CS106 & CS107) lectures focus on assignments
- CS111 lectures focus on operating systems principles and concepts
  - Sections focus on the assignments
  - Exams focus on lecture material
  - Similar to upper-division CS systems area courses

Course Information

- Website: cs111.stanford.edu
  - Calendar, Course Policies, Lecture Slides, Assignments, Gradebook (Assignment grades)
- Canvas: https://canvas.stanford.edu/courses/225586/
  Links to:
  - Panopto Course Videos - Lecture videos
  - Ed Discussion: https://edstem.org/us/courses/96128/
  - Gradescope (Exam grades)

Grading

- 55% Exams
  - 35% Final Exam
  - 20% Midterm Exam
- 35% Assignments
  - 9 weekly assignments
- 10% Participation
  - 5% Lecture Participation
  - 5% Section Participation

Exams (55% of grade)

- Midterm exam (20%) – Thursday, May 7, 7-9PM
  - Note day/time: outside of class
- Final exam (35%) – Wednesday, June 10, 8:30AM-11:30AM
  - Note day: last day of finals
- Exams: in-person, closed-book, closed-note (except for a few pages of notes) paper-and-pencil exams
  - **Proctored** as part of the Academic Integrity Working Group pilot study
  - Study protocol: Assign seats, id checking, rules on alternatives and bathroom breaks
- See course website for special exam handling (OAE, athletics, etc.)

Weekly Assignments - due Thu 11:59PM (35% of grade)

- Week 1: Assignment 0: Welcome to CS111! **(Fri 5PM)** (50 points)
- Week 2: Assignment 1: Lambdas, Threads, and Processes (80 points)
- Week 3: Assignment 2: Synchronization (100 points)
- Week 4: Assignment 3: Thread Dispatcher (100 points)
- Week 5: Assignment 4: Implementing Locks and C.V., Trust (60 points)
- Week 6: Midterm Exam
- Week 7: Assignment 5: Memory-Mapped Encrypted Files (100 points)
- Week 8: Assignment 6: Page Replacement with the Clock Alg (100 points)
- Week 9: Assignment 7: Unix V6 File System (100 points)
- Week 10: Assignment 8: Journaling File System (60 points)

Assignments

- Late policy - Keep up! See course website
- AI

Assignment 0 - Welcome to CS111!

- Available now, **due Friday 5PM**
  - No late submissions accepted except for OAE/Head TA accommodations.
- Learning goals:
  - Review basic features of C and C++
    - Remember how to write object-oriented programs with classes, methods, and instance variables
  - Use the myth machines, use gdb debugger, `sanitycheck` and `submit` tools
  - Refamiliarize yourself with some of the basic C++ template classes, such as `std::vector`, `std::map`, and `std::unordered_map`.
- Much getting started material on myth Linux environment, tools, C++, etc.

Weekly Sections

- Weekly 50-minute in-person sections led by a CA
  - Hands-on practice in small groups with the week's assignment material
  - Offered every week (except week 5) on Friday and following Monday
- Need to sign up for a section
  - Sign up on the course website
  - Section preference submissions open ASAP
    - Not first-come first-serve
    - You may submit your preferences anytime until Thursday 4/2 at 11:59 AM

Participation (10% of the grade)

- Lecture Participation (5% of grade)
  - Attendance taken using geofenced Poll Everywhere
  - Starts with a test run on Wednesday, Lecture 2 (4/2)
- Section Participation (5% of grade)
  - Attendance taken using signup sheets

CS111ACE

- CS111ACE 1-unit CR/NC supplementary companion course to CS111
  - It's designed to provide extra support and problem-solving resources for students, particularly for those who identify as coming from underrepresented or under-resourced backgrounds.
- Entry by application; section Tue 3:00 PM - 4:50 PM at Thornton 210
- In addition to all normal CS111 requirements such as section
  - See cs111.stanford.edu for more details and the link to apply
- Applications are still open! The final deadline for applications is Friday 4/3 5pm! (for late enrollment, email Fabio)

Questions?