CS202-002 OPERATING SYSTEMS

(SORRY, NO MOVIES IN THIS CLASS)

Instructor: Aditi Panda

TAs: - Zhang Han Wang
    - Sam Frank
    - Mengxi Liu
    - Leanne Lu
    - Brayton Lordianto
    - Jeff Ma

Course Website

https://cs.nyu.edu/~apanda/classes/23pa

Should already have received CampusWire Invite

Today

☐ Goals
☐ What is an Operating System?
☐ Why Study It?
☐ How Will We Study It?
☐ Mechanics, Admin, Policies
☐ History
☐ Processes

Operating Systems
Q1. How to share hardware resources between processes?
Q2. How do programs use resources assigned to them?

Sharing

1. Virtualization mechanism: Allow many programs access
2. Isolation: Bad program cannot affect another
How Accessed

→ Abstraction, API, Data Types

1. File System
   → Virtualization
   → Isolation
   → Abstraction

   int fd = open("cool", O_RDONLY);
  pread(fd, ...);

2. Memory

   Covered in class
   Later in class
   Virtualization
   Isolation
   Abstraction
③ Scheduling

**Abstraction:**

**Virtualization:**

**Isolation:**

**Why Study (Operating) Systems**
- Bridge between hardware and programs
  - Determines how your program looks
  - What performance it gets
- Is it connected?
- Is it practical?

Thinking about say LLMs

Application evolution & hw adoption often driven by systems changes

- Ton of emerging problems that need to be solved

- Provides a deeper understanding of anything running on a computer

- How study?
Course Mechanics

Components

- Class Tu-Th.
  - Recorded live on Zoom (see Brightspace)
  - Strongly encourage attending
  - Please ask & answer questions
  - Provide feedback
    - Goal is to make it useful for you.

- Reading

- HW

- LABS

- Recitation/Review
  - First one 09/11 Monday 6—7pm

- Exams
  - Midterm 10/19 During Class
  - Final
Getting Information & Help

- Check Website (daily)

- Questions About Labs & Homework
  - Post on Campuswire
  - Please help answer when possible

- Sensitive/Admin Questions
  cs202-fa23-staff@nyu.edu

- Office Hours
  - Schedule on webpage

Grading

- HW 5%
- Labs 25%
- Midterm 25%
- Final 45%

Policies

- All work that you hand in must be yours.
- Code

- What is allowed

→ Copilot & its ilk

→ Bottom line

**History**

- Batch Processing (starting 50s/60s)

- Time Sharing (coined 1950s, built 58-...)

  → Compatible Time Sharing

  → TSSS

  → MULTICS

- Unix (1969, AT&T Bell Labs)
Unix is multi-user, multi-process.

→ PORTABLE: mostly written in C starting version 4 (Nov 1973)

(first version with most things associated with Unix/
"The Unix Philosophy")

Targets minicomputers

- Parallel: development of personal computers
  → Hobbyist ➔ Apple ➔ C64 ➔ ...
  → For "professionals": Xerox PARC ➔ ...

- Merge late 1980s

→ PCs became more powerful, cheaper
Workstations replace minicomputers

- Berkeley 1980-1992
  - Tools for Unix Version 5 on
    - Vi, csh,
  - Improved kernel
  - Ported to PC
  - Eventually ends up in
    OSX

- Linux (early 1990s)
  - Written from scratch

- Lots of undiscovered paths
  - Lisp Machines
  - Plan 9
  - NT
  - Dartmouth

Processes

```c
int n = fork();
```