CS202-002: Operating Systems

- cs.nyu.edu/~apanda/classes/25fa

Instructor: Aurojit Panda

TAs: - Sarthak Khandelwal
- Hao
- Max Tang

Do this ASAP:

- Join Campuswire (go to webpage)
- Setup your laptop for class.

Today

Class Goals

Why Operating Systems

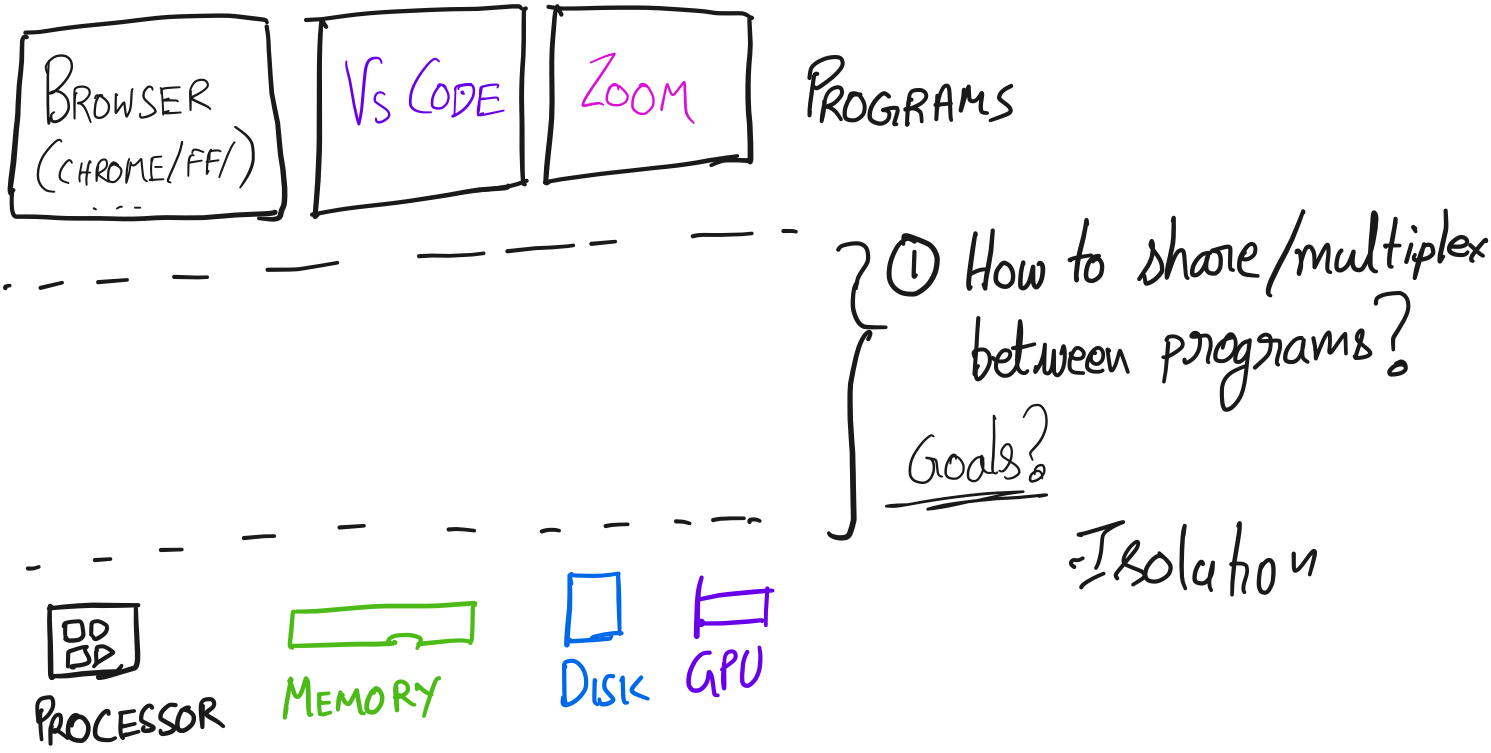
Course Mechanics, Admin, Policies

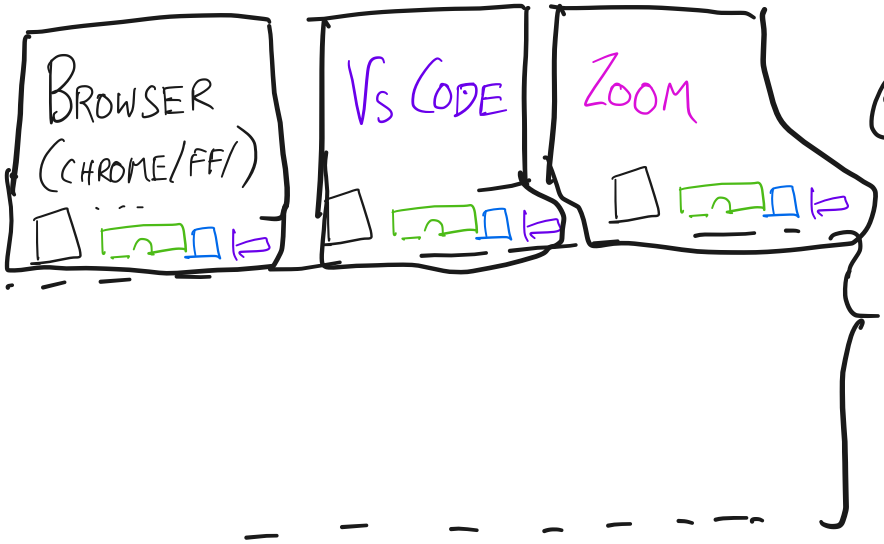
History

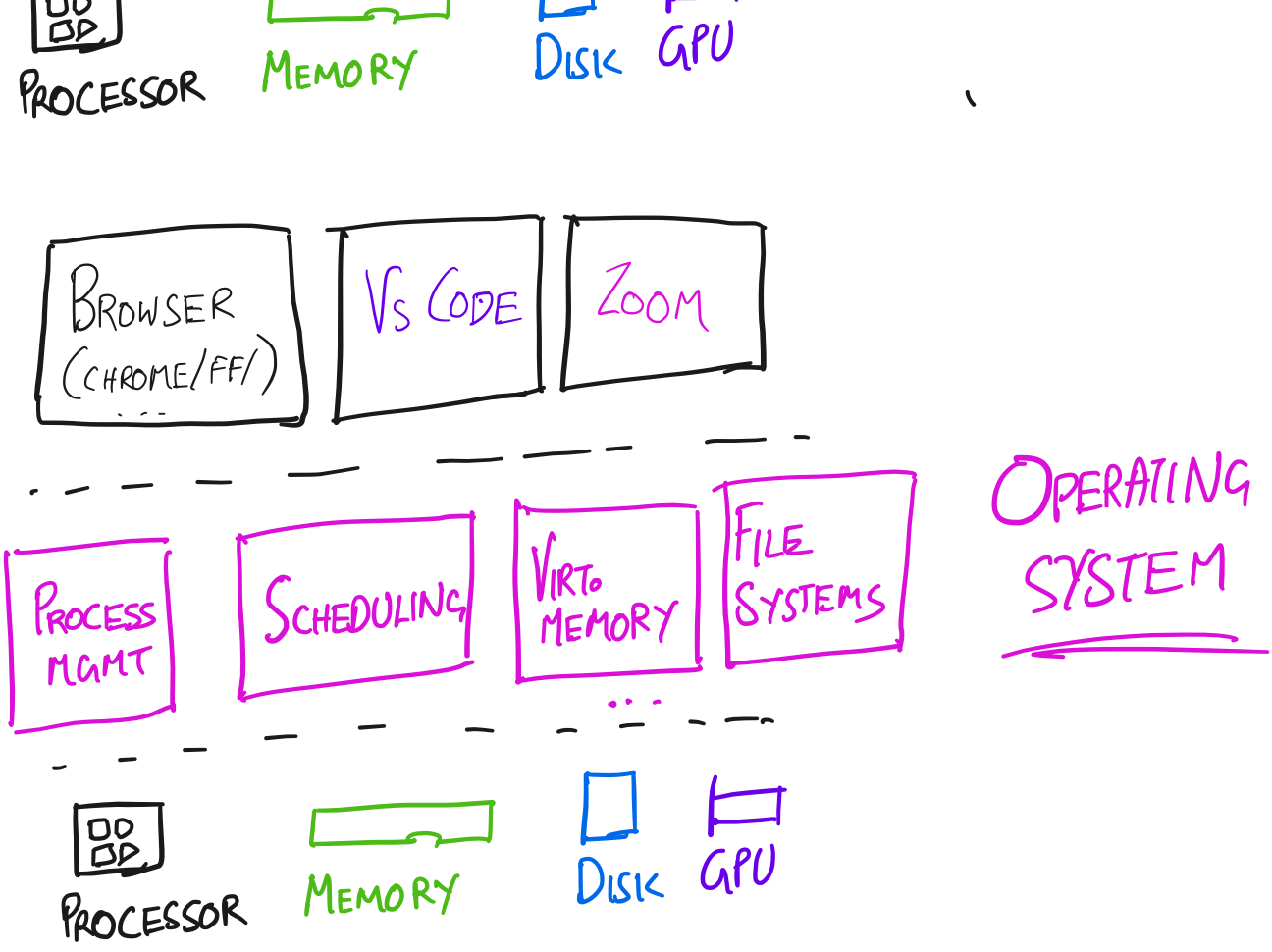
Processes

GOALS

- LEARN HOW SYSTEMS, ESP. OS WORK
- LEARN WHAT ABSTRACTIONS THEY PROVIDE
- UNDERSTAND HOW THIS AFFECTS EVERYTHING ELSE.

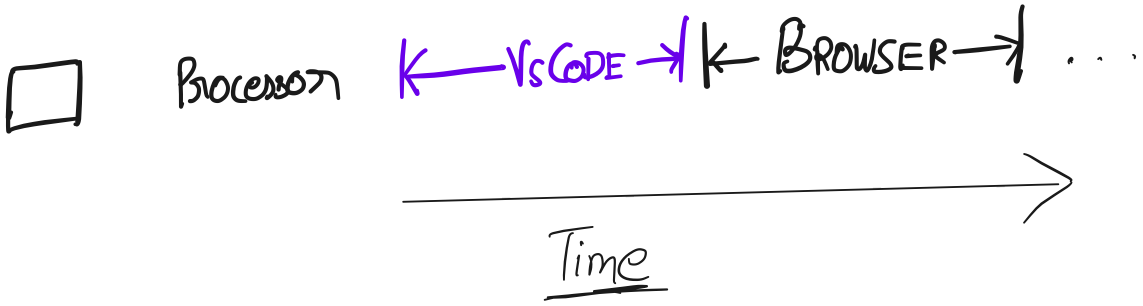






Scheduling/Process Mgmt

- Abstraction
  - Type : process
  - Interface: create (fork), destroy (kill), ...
- Sharing
  - Scheduling/multiplexing

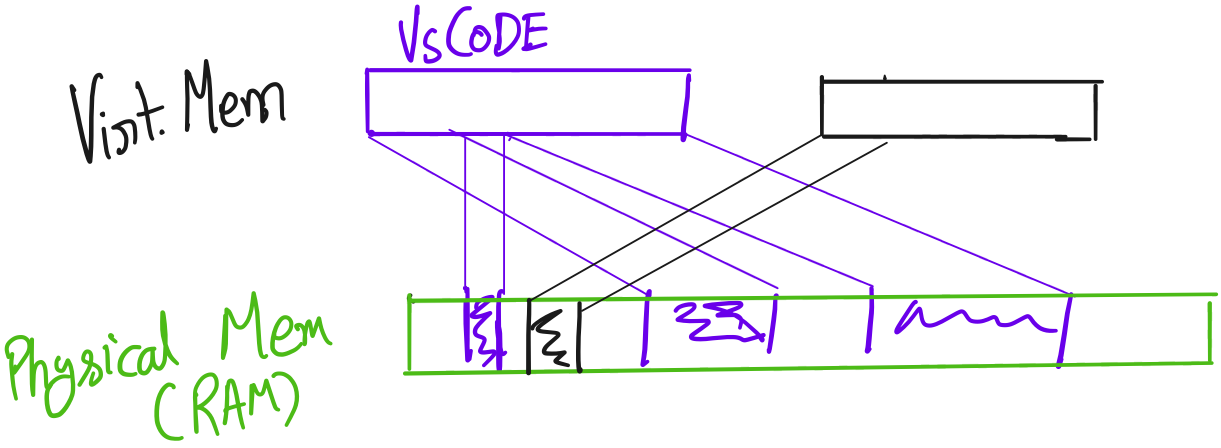


Memory

- Abstraction
  Type: Virtual memory (process)

  Interfaces: Grow (sbrk, mmap), Shrink (munmap)

- Sharing



File systems

Abstraction

Files, Directories, ...

Why study OS?

- Impact correctness, performance of every program.
  - Why is reading 1GB taking forever?
  - Why do I keep losing data in my script?
  - Why can't I allocate enough memory for this dataset.

- Understanding OS helps debug these & other problems

- Need to develop new abstractions & sharing mechs. for new problems.

  - Energy
    - Smart Phones, Laptops
    - Servers (in Datacenters)
  - Sharing GPUs
  - ...

- Develop an appreciation for what makes abstractions good/bad/ugly.

COURSE MECHANICS

- Lectures
  - Tuesday, Thursday
  - Not recorded
  - Will post
    - Notes
    - Whiteboard stuff

- Readings
  - Do them before class
    - Helps prepare you for the lectures

- Quizzes
  - 10 throughout the semester
    (See webpage for schedule)
  - 10 minutes at the beginning of class
  - Focus on material from
    - Recent lectures
    - Recent readings
  - But everything is in scope.
  - 20% of grade
    ↳ Will drop lowest 2 scores.

- Labs
  - 5 throughout semester
  - 15% of grade
  - Important for understanding course material.
  - LAB 1 is out now.

- Recitation Sections: Announce Soon!

- Exams
  - Midterm (25%) Oct 23
    During class
  - Final Exam (40%)

Getting information & help

- Check webpage for updates (Daily)

- Asking Questions
  - Campuswire
    - If posting code make your post PRIVATE
  - Admin/sensetive questions

    Email aparda@cs.nyu.edu
  - Office Hours
    - Will post a schedule shortly

- Until then: ... things up.

- Working on labs
  - Follow setup instructions linked from webpage ASAP
  - Lab 1 is out - review things from CS 201.

Policies

- All the work you turn in must be yours
  + No collaboration on labs
    - Write your own code
    - Do not show your code to anyone else
    - Do not copy code you find online.
  + No collaboration on quizzes or exams

- AI: Copilot, Cursor, Codex, ...

- Short version: Do NOT USE

- Longer version
  - Yes, they can probably do the work
  - But, that doesn't help you learn the material.
  - Not helpful in the long run

- Course difficulty

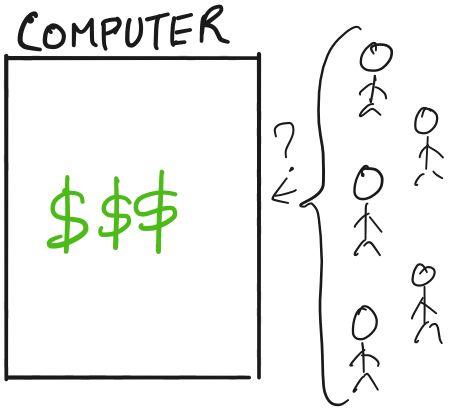
History

Most of this class is based on abstractions developed as a part of UNIX by Thompson & Ritchie 1969.

So historical focus based on that

[But really, there is a longer history

[Bell Heavy ... and a richer set of abstractions]



- Batch computing
  (HPC - Greene)

- Multiprocessing
  → Many instances
  Often designed for expensive computers

- Mid-1960s
  Minicomputers - cheaper (~$18,500 in 1965)
  smaller
  + DEC PDP line

1969: Thompson & Ritchie
Multiprocessing OS for PDP-7
UNIX - Assembly 9000 LoC

1973 UNIX v4 Rewritten in C
- Portable

1980s: Berkeley added tools & utils

(BSD)

Today: Abstractions are widely adopted

Being similar to UNIX (Posix compat) considered a positive

- Windows - various forms since 1990
- Mac OS X (~2000)
- ...

PROCESS

- Instance of a running program

PROCESS	KERNEL
- ABSTRACT MACHINE - Memory - CPU - ... - Running exactly one program	- Thing to which resources are allocated - Logic to create

- Interface to manipulate abstract machine & shared resources

abstract machine

(Late October, later in the class)

(09/04, 09/05)

OUR FOCUS for the next week or so

- PROCESS STATE

  Stored in

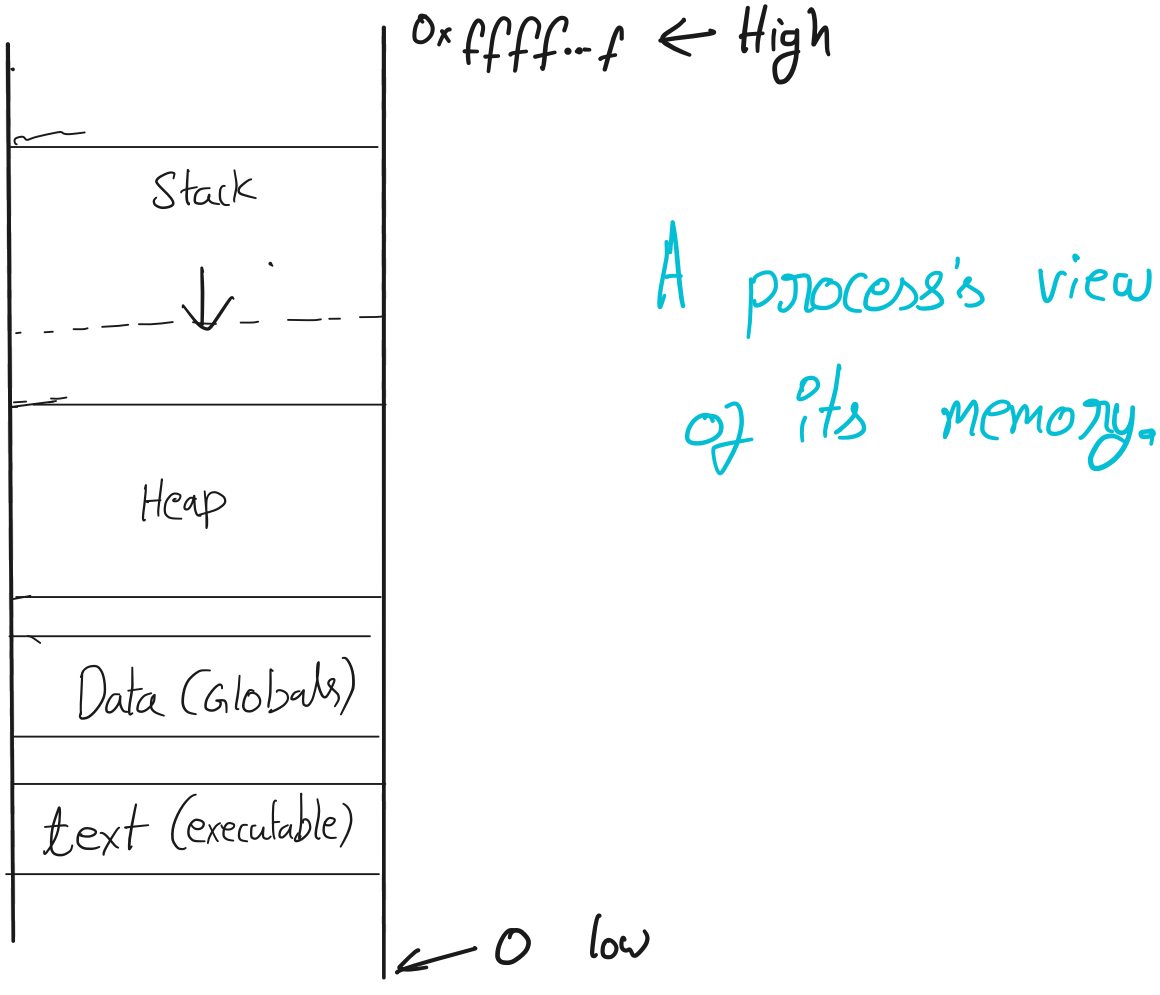
  - Registers:
    - Exclusive to the process
    - A limited number
  - Memory
    - For our purpose: exclusive to the process
    - Contains
      - The program (the thing that is running)
      - Global variables
      - Stack
        - Local variables

- "Control" info.

- Heap

- CONTROL FLOW
  - Function calls
  - Returns

} Manipulate the stack + registers



A process's view of its memory.