COMS 352:
Introduction to Operating Systems

Meisam Mohammady
Fall 2025

Teaching Team

- Instructor: Prof. Meisam Mohammady
  - ISU email:meisam
  - Office: Atanasoff 232
  - Office hours: MWF 4:20-5:00.
- TAs
  - Jahid Hasan (jhasan): Recitations; Tue&Thu 7:45-9:00 AM (Pearson2157)
  - Ayesha Samreen (ayesha62): Office hours; Thu 3:00PM-5:00PM (Pearson0112)
  - Abdullah Ali Asif (aaasif): Office hours Tue 2:00-6:00 (Pearson0112)
  -
- Piazza

Syllabus (Some Highlights)

Please find more details from the course website at Canvas!

Course Goal

- Introduce the internal operation of operating systems.
- i.e., get to know how an OS works internally?

Textbook

- Operating Systems: Three Easy Pieces (by Remzi H. and Andrea Arpaci-Dusseau)
- Available free in digital format:

https://pages.cs.wisc.edu/~remzi/OSTEP/

Prerequisites

- COMS 321 or CPRE 381
- COMS 327 or CPRE 288
- ENGL 250

- Contact instructor for any question.

Learning Activities & Assessments

- Lectures
- Recitations (discussing homework, extra examples/demonstrations, etc.)
- Attendance not mandatory but strongly encouraged

- Exams: 1 midterm exam (March 6) + 1 final exam (Final week)
- Homework Assignments (not graded; discussed in recitations)
- Programming projects
  - Project 1 based on a Unix-like teaching OS (xv6-riscv): ~ 3 parts/steps
  - Project 2 based on Linux
  - All in C programming language

- Final grade = 50% * projects + 20% * midterm + 30% * finalExam

Late Policy

- Late by 1-24 hours: 15% penalty
- Late by 25-48 hours: 30% penalty
- Late by 49-72 hours: 45% penalty
- Late by 72+ hours: not accepted!

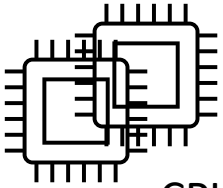
Academic Integrity

- Programming projects are individual efforts (unless specified otherwise).
- While discussion with other students is encouraged, a level of discussion that produces identical work is prohibited.
- You may not:
  - Write code with another student (or TA*)
  - Give code to another student (via email, social media, printouts, etc.)
  - Post code in a publicly accessible location
- A student found responsible for academic dishonesty will receive zero points for the work in which the dishonesty occurred. In addition, the student is subject to sanctions according to the university policy.

Introduction to Operating Systems

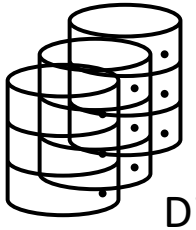
User Programs

Hardware



CPUs

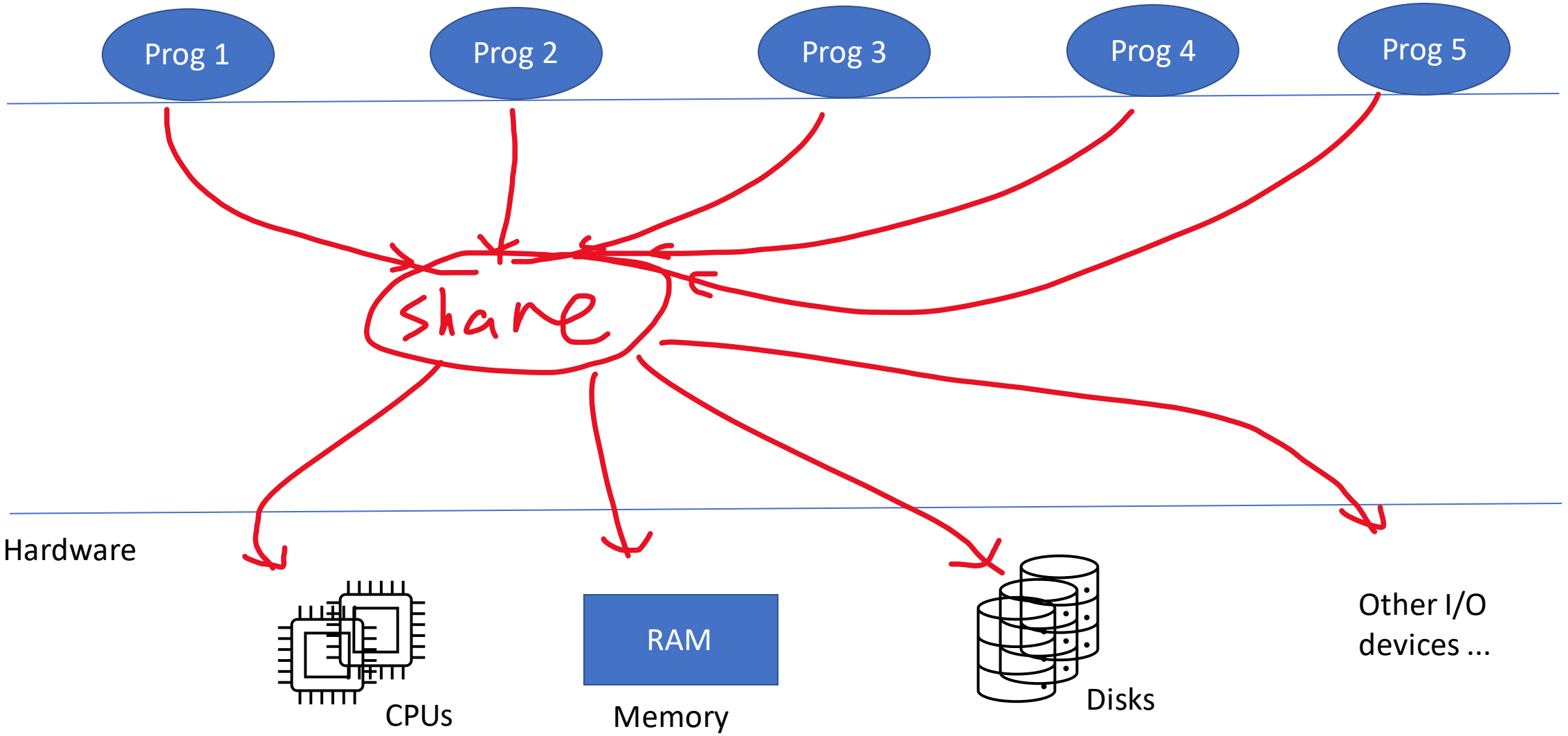
Memory



Disks

Other I/O devices ...

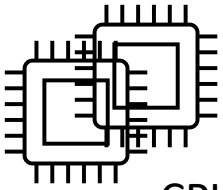
User Programs



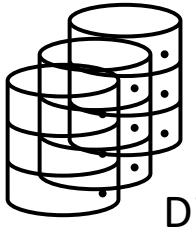
User Programs

If sharing of hardware is taken care of by the programs ==>
Problems: not easy to develop programs; conflicts in accessing resources; interfere with each other (security); code redundancy; ...

Hardware


CPUs

Memory

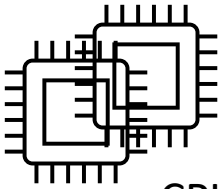

Disks

Other I/O devices ...

User Programs

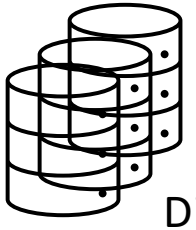
OS: Goal is to run programs more easily, efficiently, securely, …

Hardware



CPUs

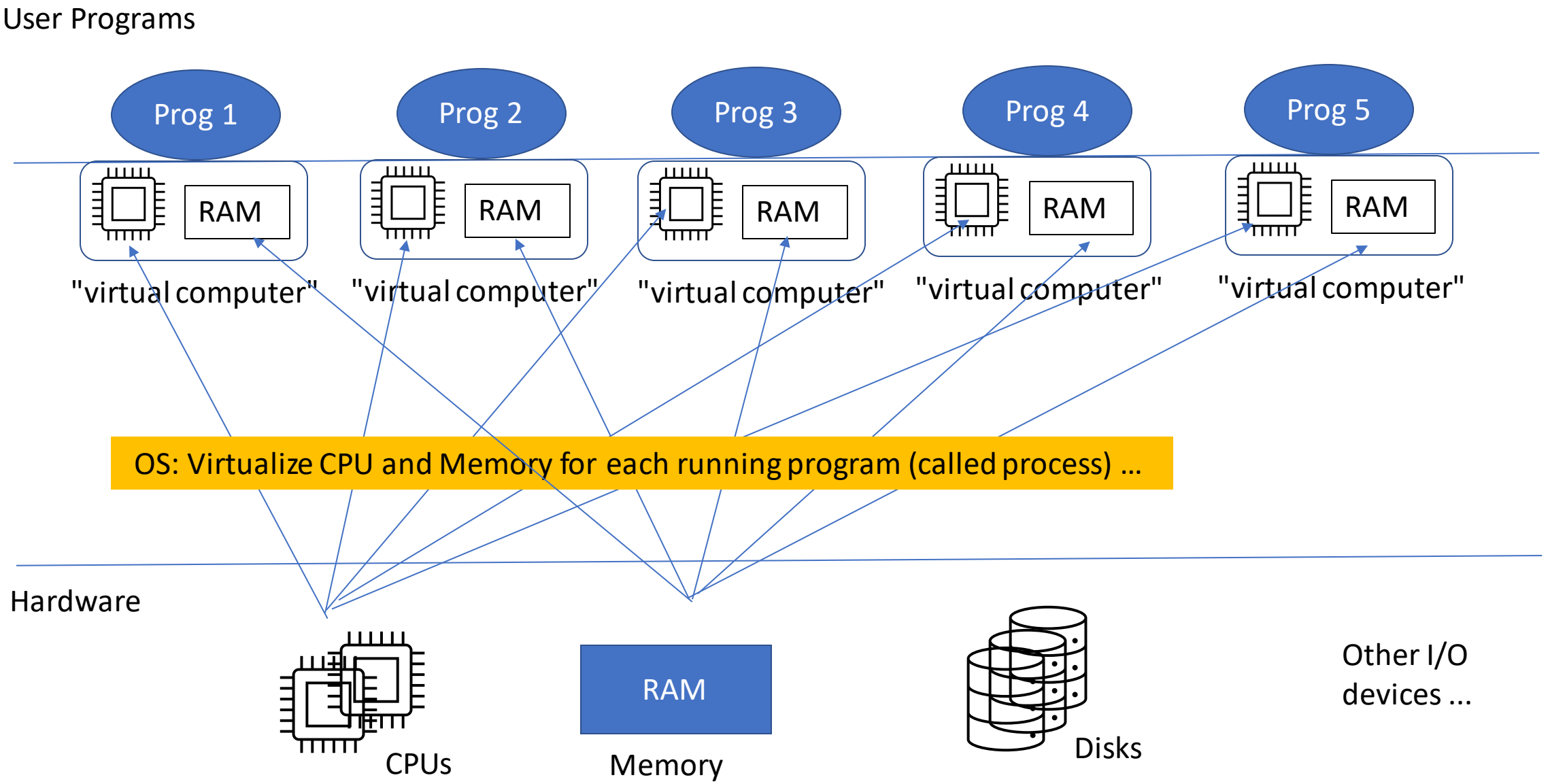
Memory



Disks

Other I/O devices ...

Goal of OS: To run programs more conveniently, efficiently, securely ….



Demo 1: Virtualizing CPU for each program

Program: cpu.c

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

int main(int argc, char *argv[]){
    if(argc != 2){ fprintf(stderr, "usage: cpu <string>\n"); exit(1);}
    char *str = argv[1];
    int i = 0;
    while(i++ < 5){
        sleep(1);
        printf("(%d) %s\n", getpid(), str);}
    return 0;
}
```

```
Prompt> gcc –o cpu cpu.c
Prompt> ./cpu A & ./cpu B & ./cpu C & ./cpu D &
```

Demo 2: Virtualizing Memory for each program

Program: mem.c

```
#include <unistd.h>
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[]){
        int p=0;
        printf("(%d) address of int variable p: %p\n", getpid(), &p);
        while(p++<5){
                sleep(1);
                printf("(%d) p: %d\n", getpid(), p);}
        return 0;
}
```

```
Prompt> gcc –o mem mem.c
Prompt> ./mem & ./mem &
```

Goal of OS: To run programs more conveniently, efficiently, securely ….

User Programs

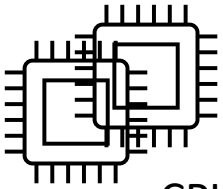
Prog 1 Prog 2 Prog 3 Prog 4 Prog 5

OS: Virtualize CPU and Memory for each running program (process)

OS: Manage concurrency in the system

- Events (in OS and programs) occur simultaneously and may interact with each other
- OS needs to:
  - hide concurrency from independent processes
  - Manage concurrency with interacting processes

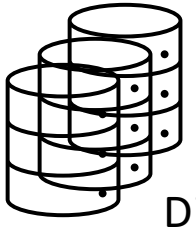
Hardware



CPUs

RAM

Memory



Disks

Other I/O devices ...

Goal of OS: To run programs more conveniently, efficiently, securely ….

User Programs

Prog 1 Prog 2 Prog 3 Prog 4 Prog 5

OS: Virtualize CPU and Memory for each running program (process)

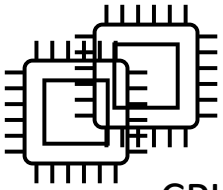
OS: Manage concurrency in the system

OS: Provide persistence

Why need?

- Lifetime of data is longer than lifetime of any process
- Machine may lose power or crash unexpectedly

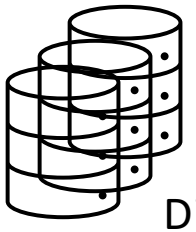
Hardware



CPUs

RAM

Memory



Disks

Other I/O devices ...

Goal of OS: To run programs more conveniently, efficiently, securely ….

User Programs

Prog 1 Prog 2 Prog 3 Prog 4 Prog 5

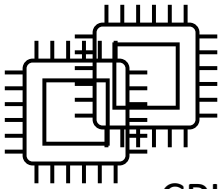
OS: Virtualize CPU and Memory for each running program (process)

OS: Manage concurrency in the system

OS: Provide persistence

OS: Support networking, system security, …

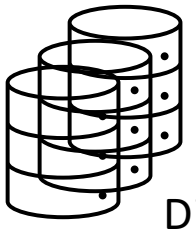
Hardware



CPUs

RAM

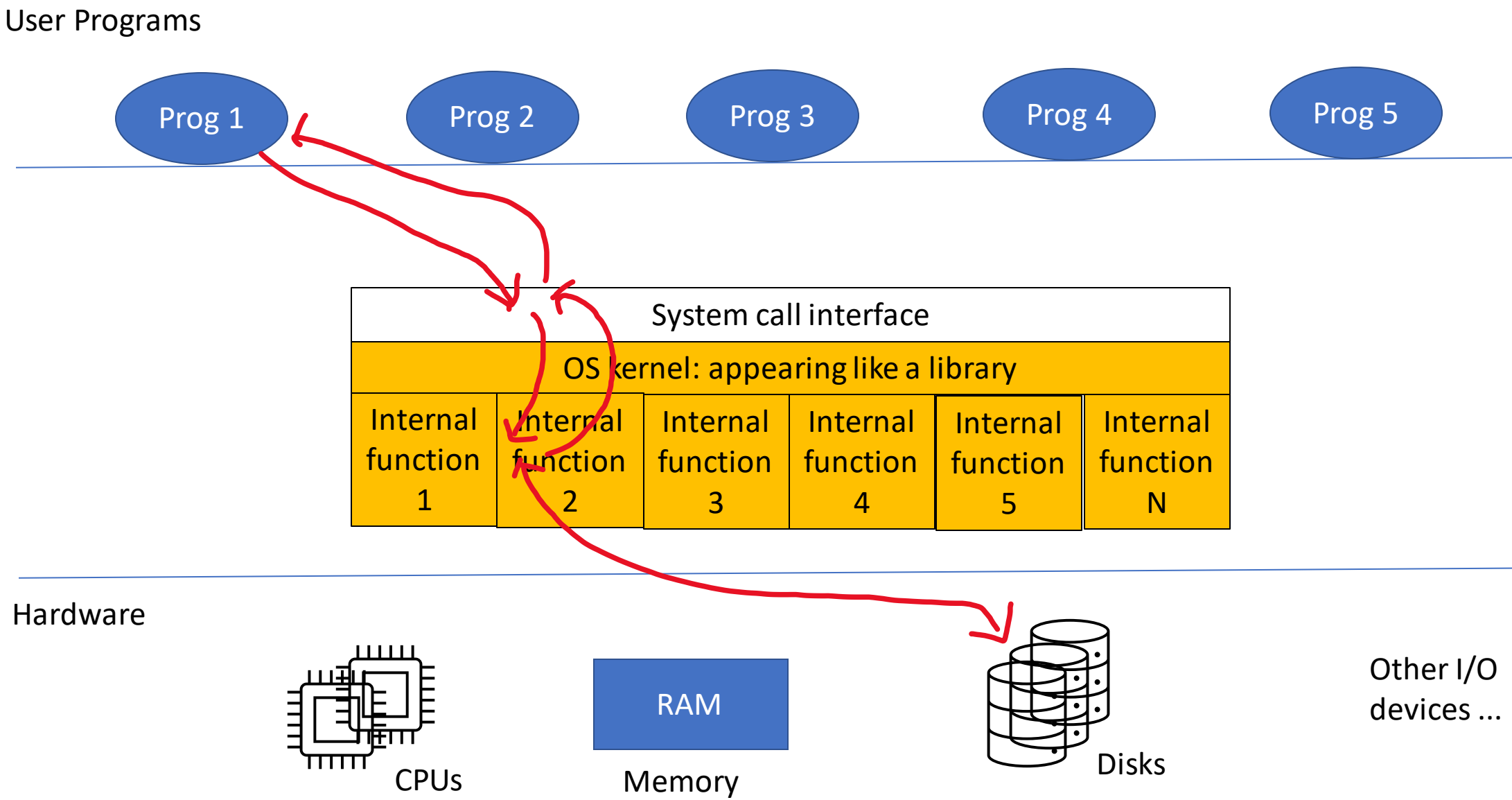
Memory



Disks

Other I/O devices ...

System Call: interface between programs and OS



Summary

We will study how an OS works based on three conceptual pieces:

- Virtualization
- Concurrency
- Persistence

We will also look into basics of networking and security

Operating systems are Complex, so programs can be Simple.

(Read Ch 2 of the textbook for more on the introduction of OS.)

Next Lecture: Process

Read Ch 4 & 5 on process abstraction and API ...