

# Operating Systems

(Sistemas Operativos)

## Introduction

University of Minho  
2024-2025



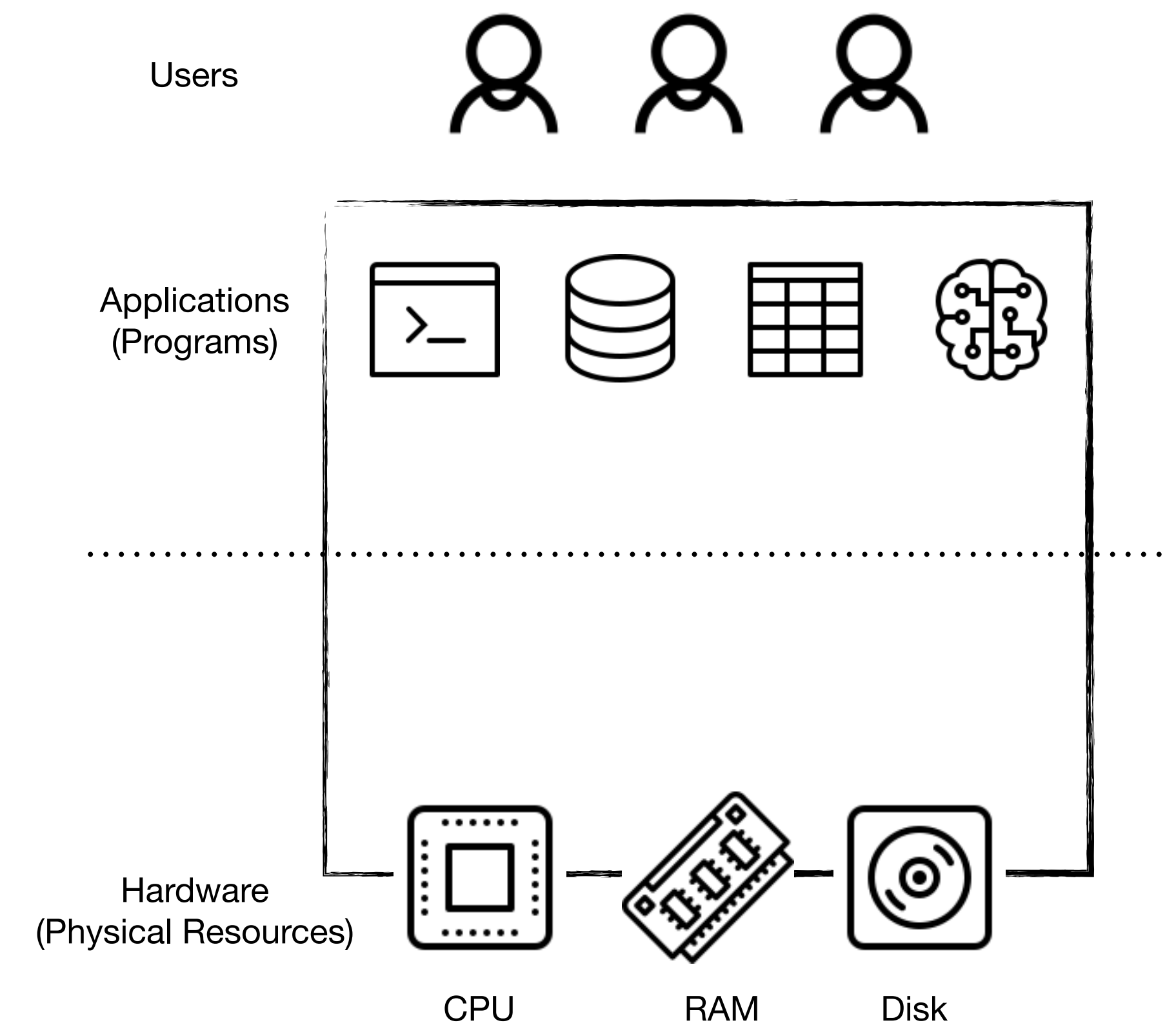
# The Landscape

## ● Applications (Software)

- Programs executing millions / billions of instructions per second (e.g., text editors, databases, AI pipelines)

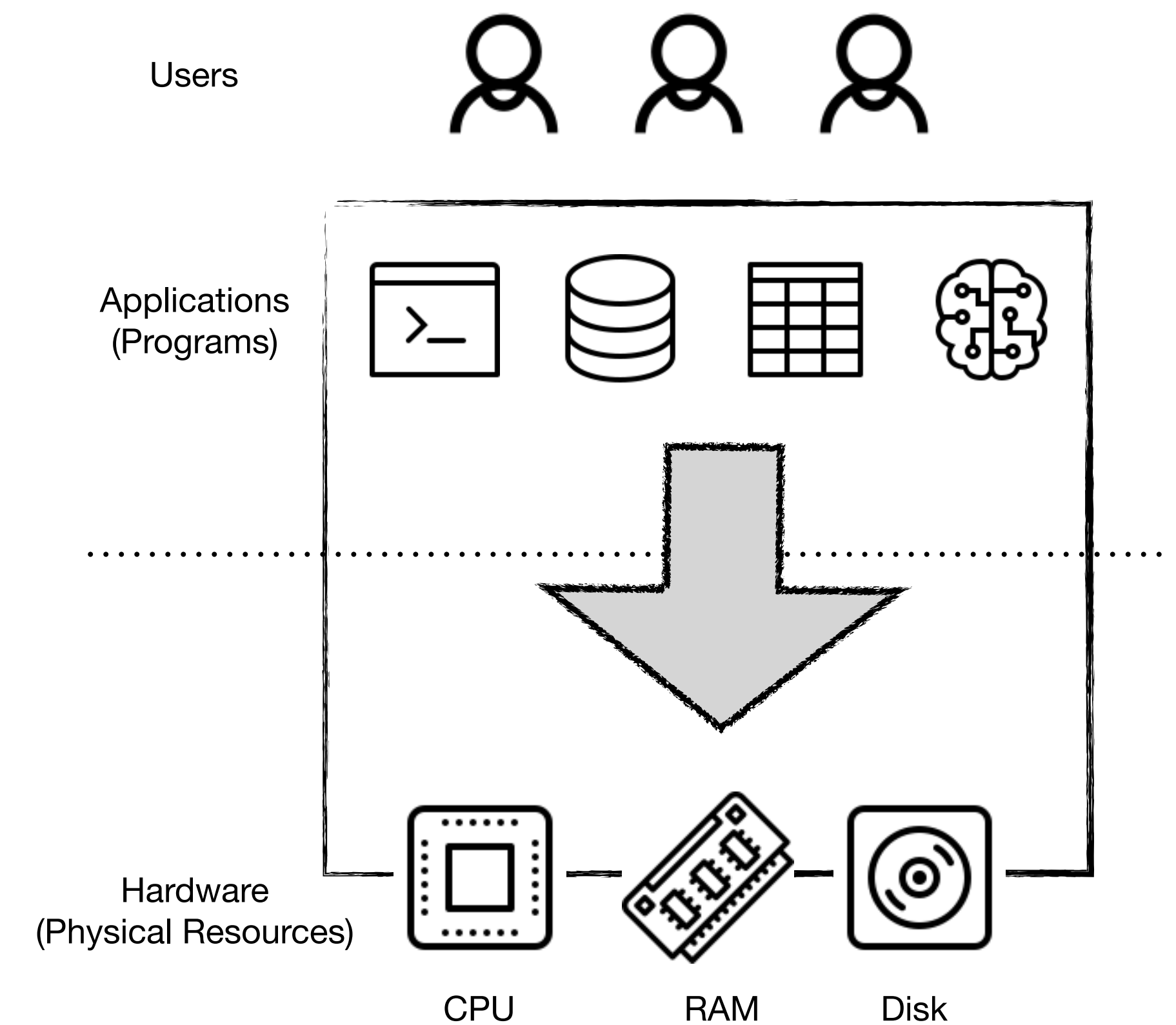
## ● Physical resources (Hardware)

- CPU: where instructions are executed
- Memory: stores volatile data structures
- I/O Devices: external devices to interact with
  - Disk: to store data persistently
  - Keyboard, mouse, display, printer, ...



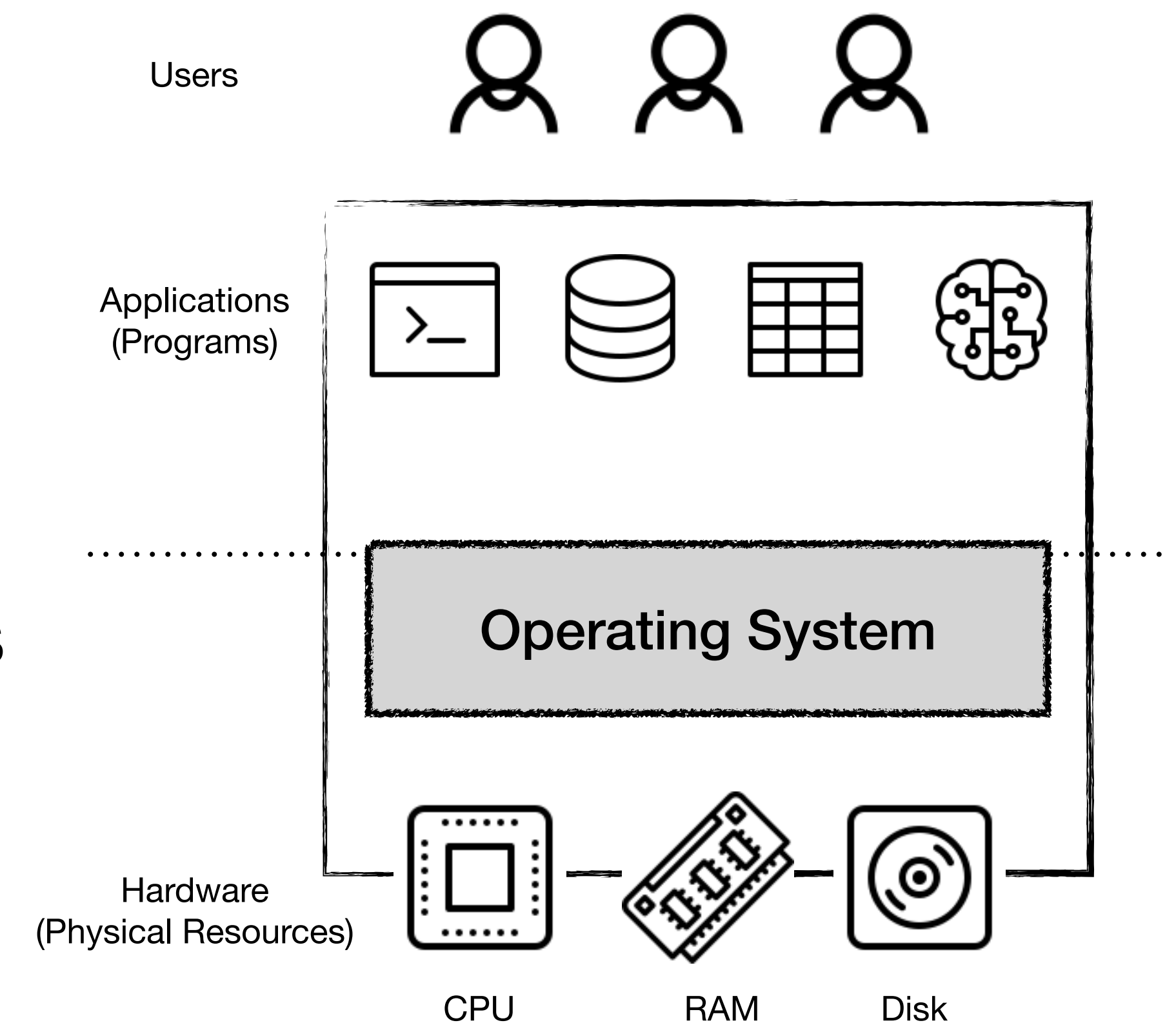
# The Challenges

- Writing code that uses directly the hardware interfaces is challenging...
  - Different hardware devices and interfaces
  - Low level languages (e.g., Assembly)
  - Time consuming and error-prone
- and requires addressing several concerns
  - Efficient usage of hardware resources
  - Resource sharing (e.g., CPU, memory, disk) across multiple running programs
  - Security and isolation of running programs
  - ...



# Enters the Operating System (OS)

- Mediates the interaction between programs and hardware
- **Abstracts** physical resources into general, powerful, and easier-to-use virtual forms
- Offers **interfaces** (APIs) to use these resources (e.g., run programs, use memory and devices)
- **Manages** resources for multiple programs (e.g., concurrent programs, resource sharing)



# Why Should I Do This Course?

- The OS is a powerful ally to efficiently run multiple programs on a computer, however:
- The incorrect use of OS tools and APIs may lead to
  - Inefficient programs (bad performance)
  - Incorrect programs (bugs, crashes, data corruption)
- To correctly use these tools and APIs one must:
  - Try them - **practical classes!**
  - Understand them (means knowing the OS) - **theoretical classes!**
- And finally, because **Operating Systems are Fun!**
  - You will study algorithms, data structures and optimizations applicable to different contexts (AI pipelines, databases, web services, apps, ...)



# Roadmap

## Theory

- Introduction to Operating Systems
  - Design goal(s) and evolution of OSs
  - Modern OS stack, tools and interfaces
- CPU Management
  - The process abstraction and API
  - CPU scheduling
- Memory Management
  - Address space management
  - Extending physical memory
- Persistence
  - I/O devices (HDDs, SSDs, RAIDs)
  - File Systems interface and design

## Practice

- Persistence API
  - Creating, writing, and reading files
  - Improving programs' performance when accessing and persisting data
- Processes API
  - Creating processes
  - Sequential and concurrent programs
  - Executing programs
- Process Communication API
  - Efficiently exchanging data across
    - related processes
    - un-related processes

# Assessment

- Project (50%)

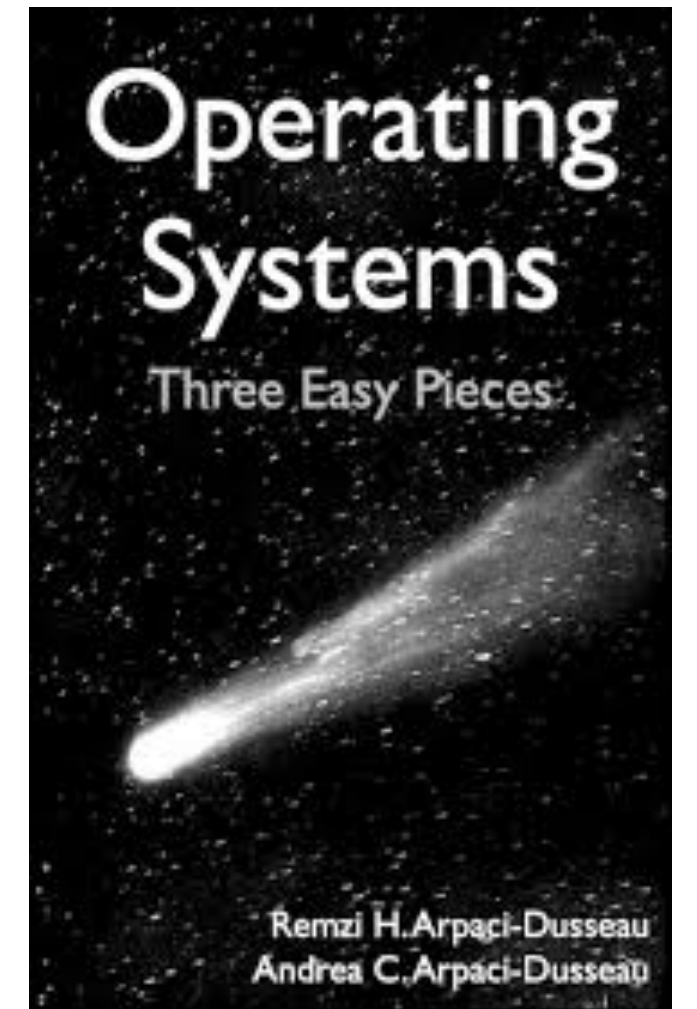
- Report: 17/05/2025
- Presentation: 02/06/2025 - 05/06/2025

- Written exam (50%) – minimum grade: 8 values

- 27/05/2025

# Main References

- Remzi H. Arpaci-Dusseau, Andrea C. Arpaci-Dusseau.  
**Operating Systems: Three Easy Pieces.** Arpaci-Dusseau Books, 2018.



- Avi Silberschatz, Peter Baer Galvin, Greg Gagne.  
**Operating System Concepts (10. ed).** John Wiley & Sons, 2018.





# Team (LEI + LCC)

- João Paulo (LEI+LCC) - jtpaulo@di.uminho.pt
- Ana Nunes (LEI) - ananunes@di.uminho.pt
- Victor Fonte (LEI) - vff@di.uminho.pt
- Pedro Peixoto (LEI) - D14110@di.uminho.pt
- Tânia Esteves (LEI+LCC) - d12729@di.uminho.pt
- Ricardo Macedo (LCC) - d12010@di.uminho.pt
- Francisco Neves (LEI) - D8328@di.uminho.pt
- Rúben Daniel Adão (LEI) - ruben.d.adao@inesctec.pt

# Questions?