# Operating Systems Concepts

CS 4375, Fall *2025*
**Instructor:** MD Armanuzzaman (***Arman***)
*marmanuzzaman@utep.edu*
August 25, 2025

# Agenda

- Course Overview
- Instructor and TA
- Course logistics
- Syllabus
- Learning outcomes
- Course motivation
- Introduction to OS

# Course Description

- CS 4375 is a course about the **systems-level software** called the **operating system** that provides an interface between **application software** and the **computer hardware**.
- The operating system is **responsible** for sharing **resources**, providing common **services**, and **protecting** programs from interference from other programs.
- Topics covered in the course include **process management** and **scheduling**, **concurrency**, **interprocess** communication, **memory** management, **input/output** management, **file systems**, and **networking** basics.

# Instructor

- MD Armanuzzaman (*Arman*)
  - He/him/his
- Research Area
  - Systems and Software Security
- PhD: University at Buffalo
- Email: *marmanuzzaman@utep.edu*
- Room: CCSB 3.1008
- Office Hours: ***M***, *1.30 PM - 3.00 PM* Or *by appointment*

# Teaching Assistant

- Name: Daniel J Marin
- Email: *djmarin1@miners.utep.edu*
- Room: CCSB 1.0706
- Office Hours: ***WF***, *3.00 PM - 5.00 PM* Or *by appointment*

# Prerequisites

- Basic data structures
- C programming
  - Pointers
- If you do run into a issue that you don't have a background, it is **your responsibility** to take appropriate steps to complete your understanding
  - Don't give up
  - Read the reference materials
  - Go to office hours
  - **Ask questions**

# Class policies

- Do your own work
- No cheating
- No unauthorized collaboration
- Do not violate the academic integrity policy
- Be respectful

# Logistics

- Class website:
  - Schedule, Syllabus, and slides (after lectures)
- Blackboard
  - Quiz, Syllabus, and Assignment submissions
- Microsoft teams (*Invited*)
  - Assignments and Communication
- Github
  - Hosting assignment solutions
- No videos of the class will be provided
  - Attendance will be graded

# Logistics

- Assignment are due at **11.59 PM**
- No extensions will be allowed without proper justification
- Exams will be in person
- Generative AI usage:
  - Discovering algorithms
  - Generating code for programming assignments
  - MUST DISCLOSE ASSISTANCE
  - ***Office of Student Conduct and Conflict Resolution***

# Grading Structure

- 500 points: **Homework assignments**
- 100 points: **Quizzes**
- 150 points: **Midterm exam**
- 250 points: **Final exam**

A = 1000-900 || B = 899-800 || C =799-700 || D = 699-600 || F = 599 and Below

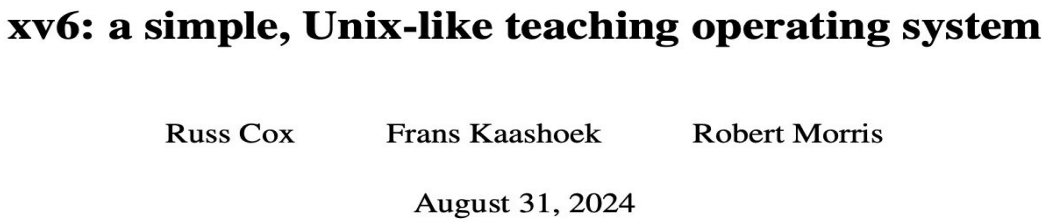
# Books/Materials

- **Operating Systems: Three Easy Pieces, by Remzi H. Arpaci Dusseau and Andrea C. Arpaci-Dusseau**
- **xv6: a simple, Unix-like teaching operating system, by Russ Cox, Frans Kaashoek, and Robert Morris. RISC-V version**
- The C Programming Language, 2nd edition, by Brian Kernighan and Dennis Ritchie
- The Linux Programming Interface, by Michael Kerrisk

**I can find out materials, once I know what I need to learn**

# Homeworks/programming assignments

- Six programming individual assignments
- We will use ***xv6*** teaching operating systems from MIT
  - *RISC-V 32-bit version*
- Assigned on Teams and turned in using GitHub
- **Most of you learning will come from the homeworks**

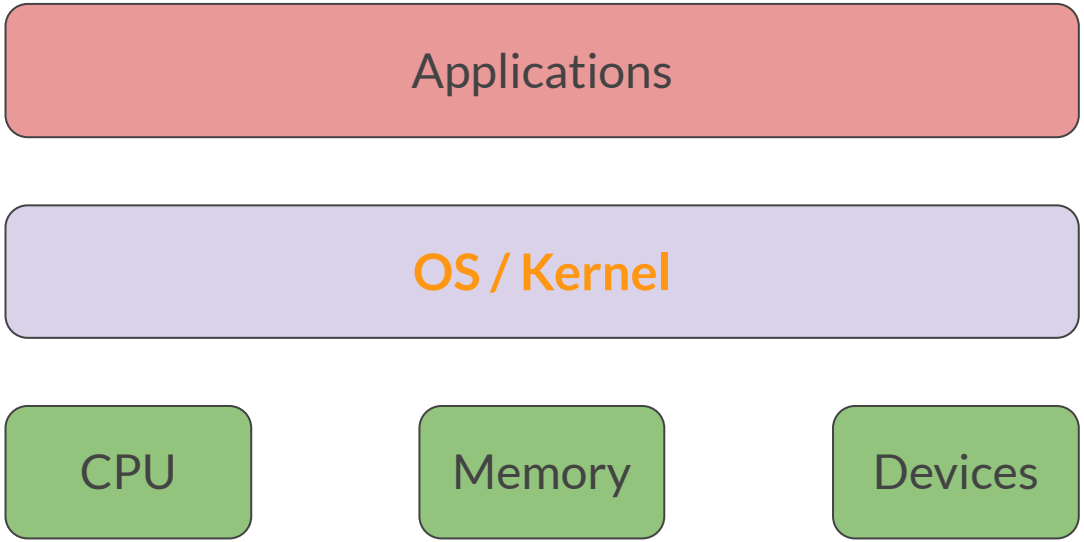


**xv6: a simple, Unix-like teaching operating system**

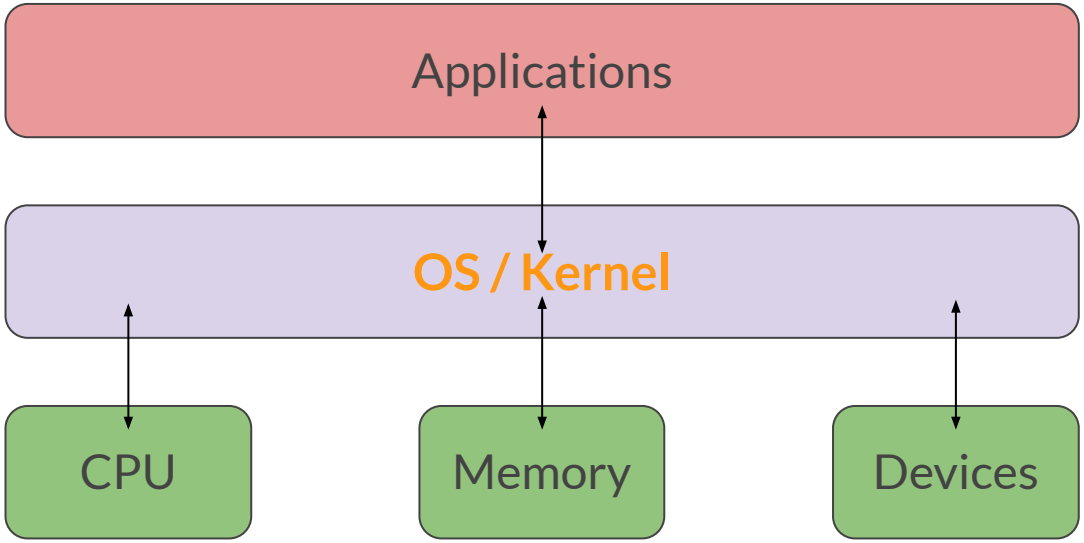
Russ Cox Frans Kaashoek Robert Morris

August 31, 2024

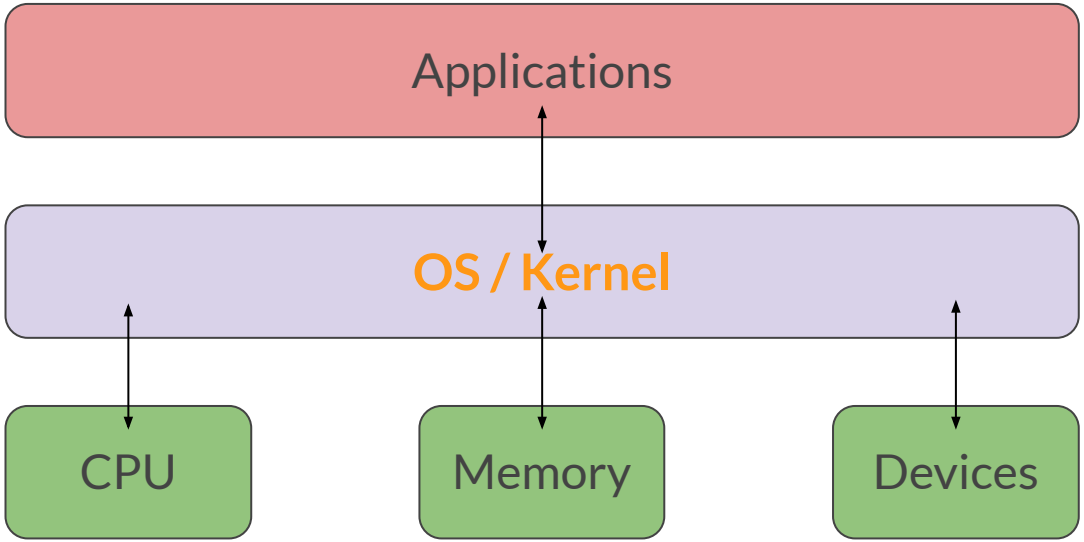
# Course motivation



# Course motivation



# Course motivation

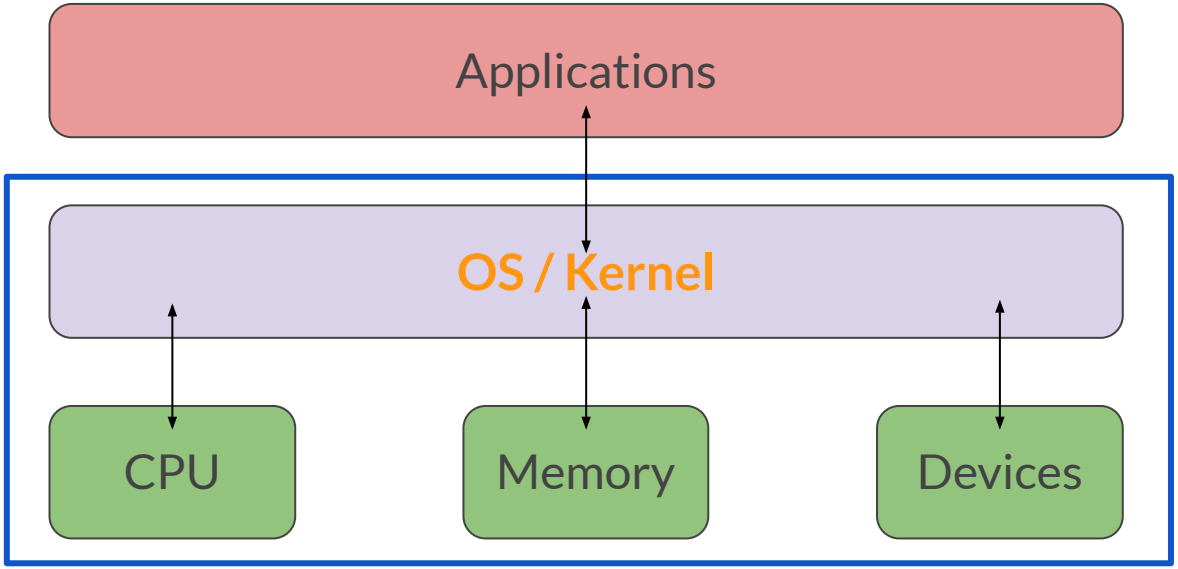



Everything you do as a software engineering relies heavily in OS-level efficiency

# Course motivation

- OS is the “magic” that provides abstractions for CPUs, devices, address space, and network communication
- Performance, functionality, security, and reliability
- Environment in which your application executes
- Troubleshoot problems that come from outside your application
- **Industry demand for people with solid systems background**
- Exciting research and development opportunities in IoT, cloud and edge computing, support for specialized architectures

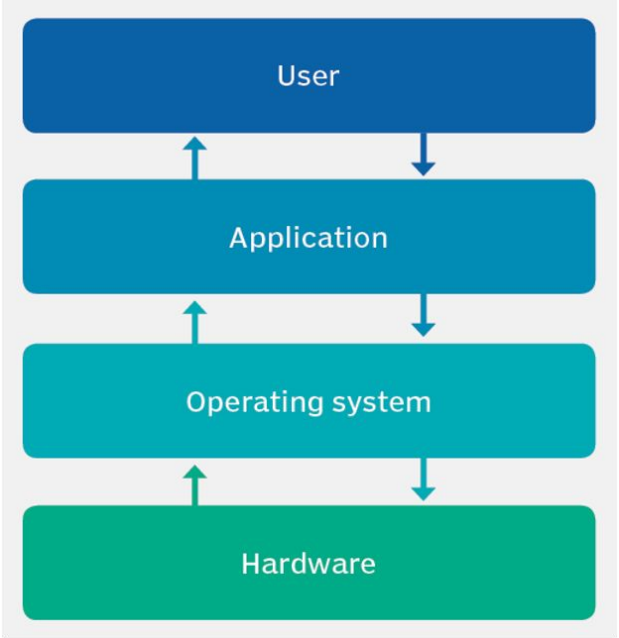
# My Research: Systems & Software Security




*Embedded, Mobile, Desktop, Distributed systems*

# What is an OS and what does it do?

- An operating system (OS) is the systems software that **interfaces** application programs with the underlying hardware
- Applications make requests for services through a **defined API**s
- The OS **manages** and **interfaces** to underlying hardware (e.g., **processors, memory, storage devices, network interfaces**) so that applications don't need to know about hardware details
- The OS **launches** and **manages** every application, including multiple processes or threads.



# Types of Operating Systems

- General-purpose operating system
  - Run multiple applications in broad range of hardware
  - Windows, MacOS, Linux
- Mobile operating system
  - Efficient performance and resource usage and fast response time
  - Apple iOS, Google Android
- Embedded operating system
  - Usually provided on a chip that is incorporated into the device
  - ATMs, IoT devices, medical devices
  - Embedded Linux
- Real-time operating system
  - Respond quickly and predictably under time constraints
  - FreeRTOS, zephyr




# Tasks

- Go over the syllabus carefully
- Take a look a to the tentative schedule
- Sharpen/improve your expertise with C programming
    - Pointers
- Refer to materials in the schedule

# Thank you