CS 2080: PROGRAMMING WITH UNIX

DR. ARMIN MOIN, ASSISTANT PROFESSOR

QUANTUM-CLASSICAL AI AND SOFTWARE ENGINEERING (QAS) LAB

DEPARTMENT OF COMPUTER SCIENCE

UNIVERSITY OF COLORADO COLORADO SPRINGS (UCCS)

SUMMER 2024

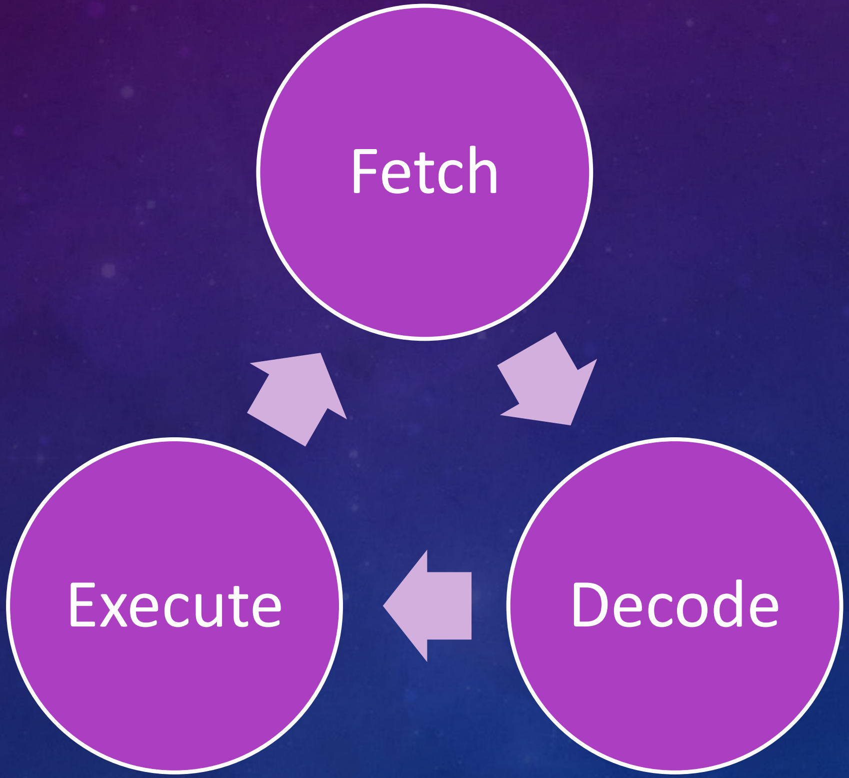
LECTURE 2: BASIC OPERATING SYSTEM CONCEPTS

- Outline:
  - What is an OS?
  - Computer architecture and organization
  - OS components
  - Interrupts (hardware and software-based)
  - Processes and multitasking
  - Dual/multi modes of operation
  - OS timer
  - Principle-based Ethics
  - Team formation and project ideation revisited

OPERATING SYSTEM (OS)

- A software system that manages a computer's hardware resources and allocates those resources to programs.
- Hardware resources: E.g., CPU, memory, I/O devices, and storage.
- Examples of OSs: UNIX, other UNIX-Like OSs, such as GNU/Linux, Microsoft Windows, Apple's Mac OS, TinyOS, ContikiOS

THE CPU INSTRUCTION CYCLES



COMPUTER ARCHITECTURE AND ORGANIZATION

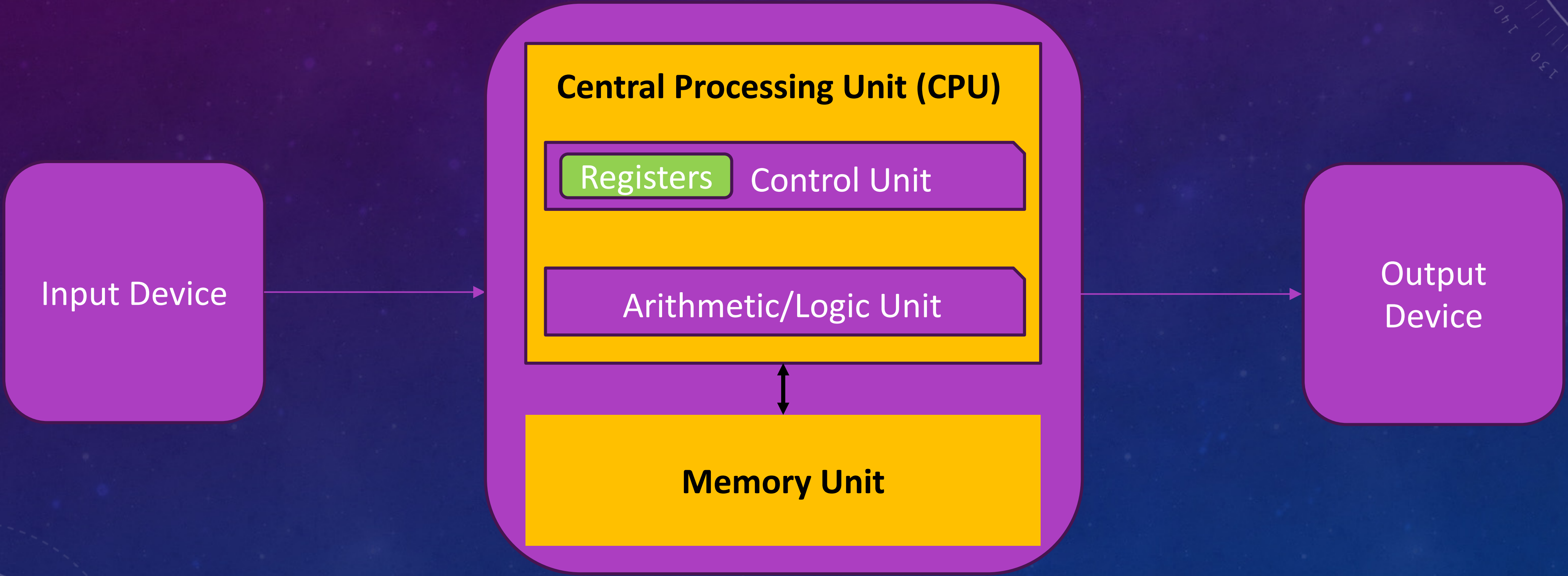


Figure inspired by Silberschatz et al. 2018

A TYPICAL PERSONAL COMPUTER (PC) SYSTEM

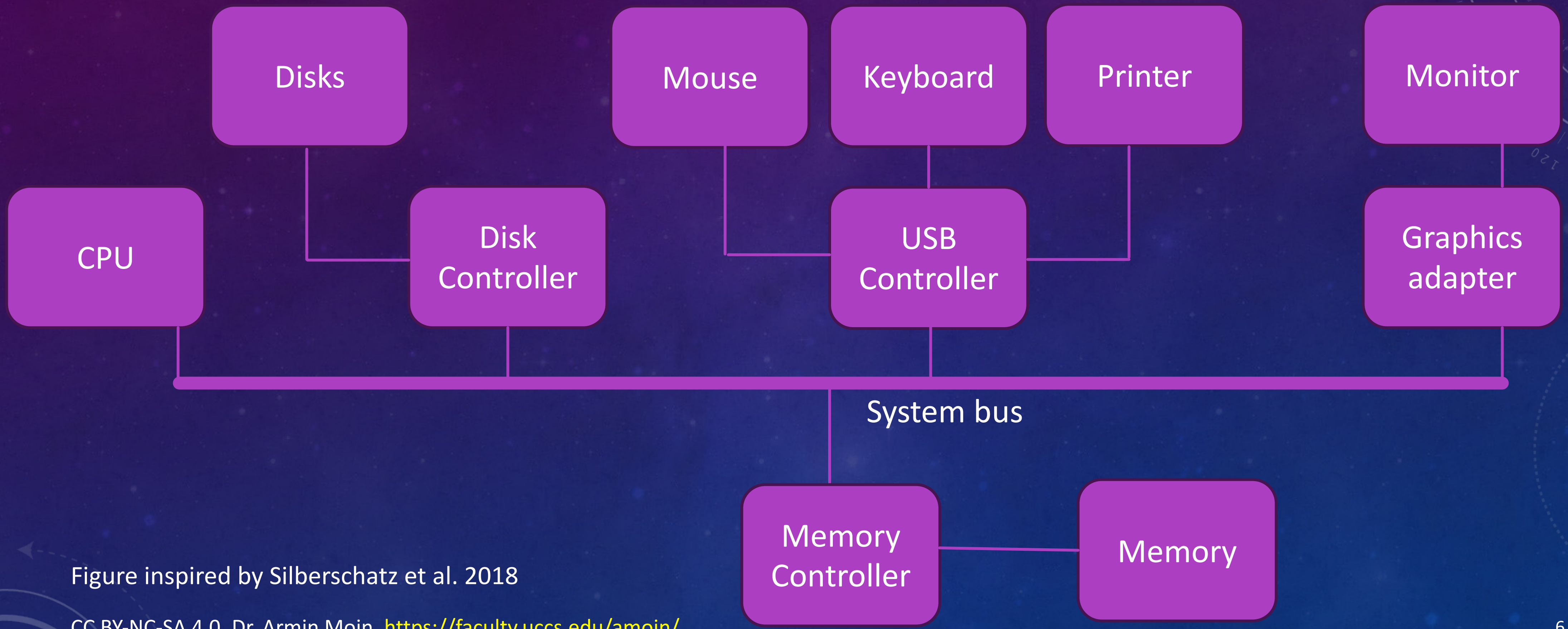
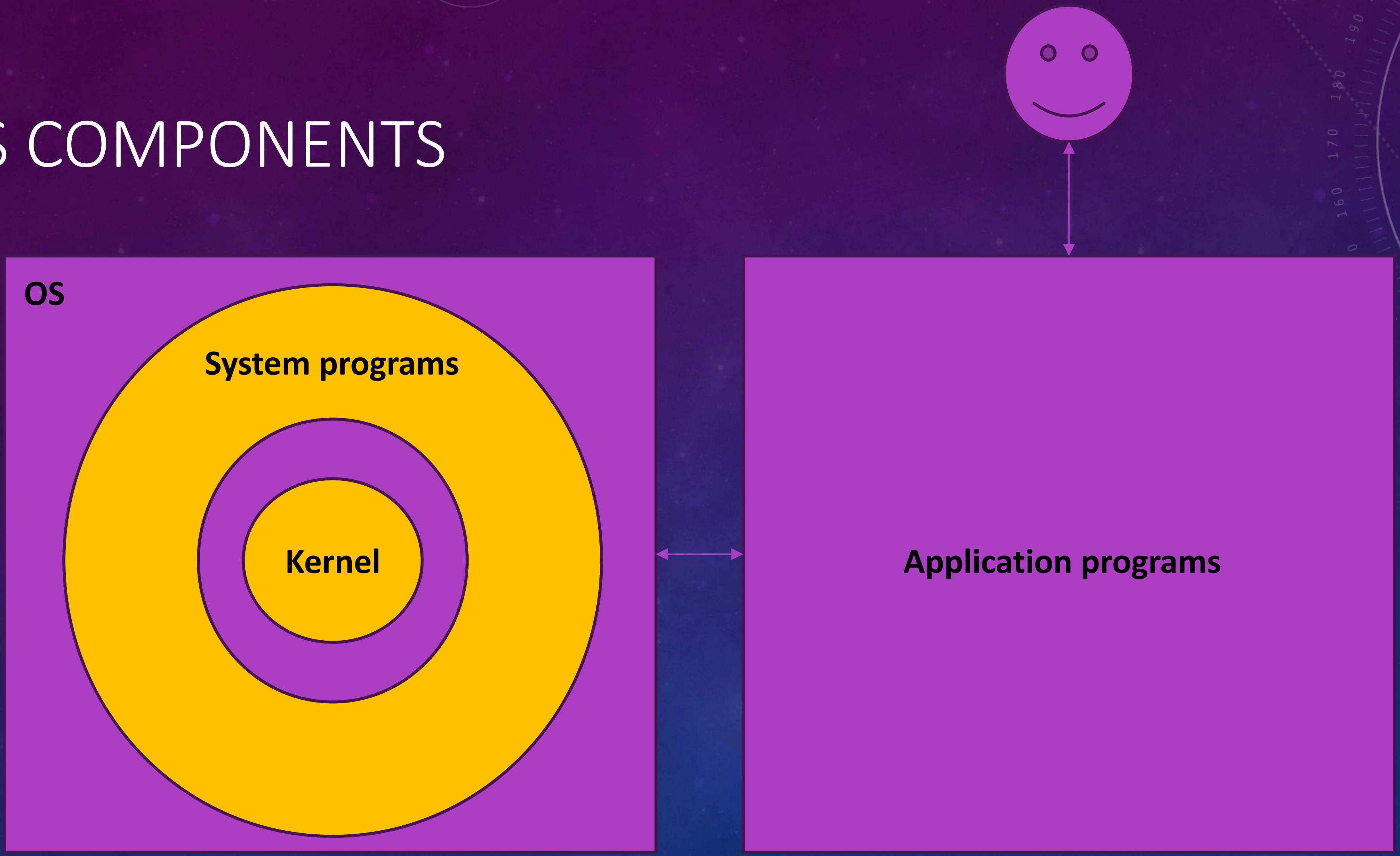


Figure inspired by Silberschatz et al. 2018

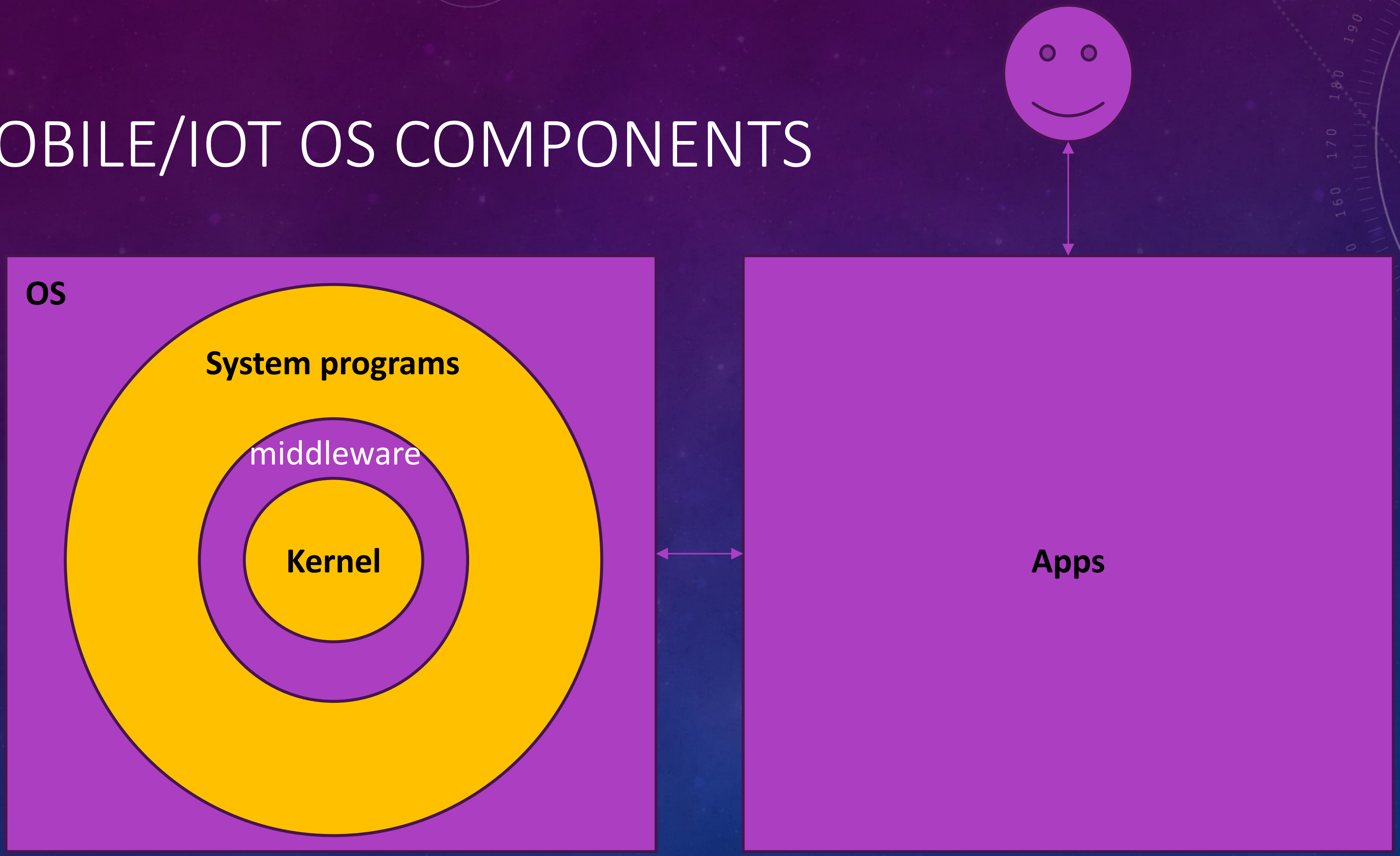
DEVICE DRIVER

- An OS has typically a device driver for each device controller, which offers a uniform interface.
- The CPU and the device controllers run in parallel and compete for memory cycles.
- Memory controller: Orderly access to shared memory

OS COMPONENTS



MOBILE/IOT OS COMPONENTS



RESOURCE-CONSTRAINED IOT DEVICES

- Most of the resource-constrained Internet of Things (IoT) devices have no OS!

BOOTSTRAP

- A small program stored within the computer hardware in firmware (e.g., BIOS / UEFI)
- Initializes all aspects of the computer system when it is powered on: E.g., CPU registers, memory, and device controllers.
- Bootstrap loads the OS kernel into the memory
- Once the kernel is loaded, it can provide services to the system and its users

BOOTSTRAP

- Some services are provided outside of the kernel by system programs loaded into the memory at boot time, known as system daemons, that run the entire time in the background.
  - E.g., On Linux, ***systemd*** is the first system program that runs many other daemons.
- The OS waits for an event, such as a hardware interrupt or a software interrupt (i.e., an exception, such as division by zero, or a system call).

THE INTERRUPTS MECHANISM

- The interaction between the OS and the hardware
- Sending a signal to the CPU via the system bus
- CPU interrupted -> stops what it is doing -> immediately transfers execution to a fixed location (i.e., the starting address of the service routine for the interrupt) -> then, the CPU resumes its previously interrupted execution

THE INTERRUPTS MECHANISM

- The device controller ***raises*** an interrupt by asserting a signal on the interrupt request line, the CPU ***catches*** the interrupt and ***dispatches*** it to the interrupt handler, and the handler ***clears*** the interrupt by servicing the device.
- Nonmaskable and maskable interrupts (the latter can be turned off during the CPU's critical processing)

MULTIPROGRAMMING AND MULTITASKING

- Multiprogramming: The ability of an OS to run multiple programs. This increases the CPU utilization.
- In a multi-programmed system, a program in execution is called a process.
- A process is the unit of work in a system.

MULTIPROGRAMMING AND MULTITASKING

- The OS keeps several processes in memory simultaneously. The CPU switches between them when it needs to wait; thus, it is never idle.
- Multi-tasking: A logical extension of multiprogramming. The CPU switches among tasks frequently. The response time is fast. This satisfies the user.
- CPU scheduling, disk storage (file system), memory management (virtual and physical), as well as process communication and synchronization need to be taken care of.

DUAL MODE AND MULTIMODE OPERATION

- Separating the execution of the OS code and the user-defined code by hardware support.
- Kernel mode (superuser mode) vs. user mode
- Mode bit: 0 (kernel), 1 (user)
- Advantage: a malicious program cannot harm other programs or the OS

DUAL MODE AND MULTIMODE OPERATION

- The concept can be extended to multiple modes: e.g., the Intel processors with 4 protection rings
    - Ring 0 = kernel, Rings 1 and 2 rarely used, Ring 3 = user
- If the CPU supports virtualization, a separate mode is needed that indicates when the Virtual Machine Manager (VMM) is in control.
    - user processes' privileges < VMM privileges < the kernel's privileges

VIRTUALIZATION VS. CONTAINERIZATION

- Virtualization: Running multiple OS instances on one machine – Hardware support
  - E.g., VirtualBox or VMware
- Containerization: Running a single OS instance on one machine with multiple user spaces to isolate processes (supports multi-tenancy) – Software support
  - E.g., Docker

THE OS KEEPS WATCHING

- Checks for illegal instruction or illegal memory access (e.g., if the memory address is not in the user's address space)
  - The OS must terminate the program abnormally
  - The same code as the user-requested abnormal termination
  - Appropriate error message given and the memory dump written to a file

THE OS TIMER

- The OS sets a timer before turning over control to the user mode.
- The timer counts down until it reaches 0. In that case, it interrupts, and the control is automatically transferred to the OS. The OS may decide to give the process more time.
- On Linux systems, the kernel configuration parameter ***HZ*** specifies the frequency of timer interrupts (i.e., the number of timer interrupts per second). Also, the kernel variable ***jiffies*** indicates the number of timer interrupts that have occurred since the system was booted.

PRINCIPLE-BASED ETHICS

- Daniels Fund Ethics Initiative
  - https://www.danielsfund.org/ethics/overview
  - https://business.uccs.edu/resources/ethics
- 8 core principles

DANIELS FUND ETHICS INITIATIVE – PRINCIPLE 1

- Integrity
  - Act with honesty in all situations

DANIELS FUND ETHICS INITIATIVE – PRINCIPLE 2

- Trust
  - Build trust in all stakeholder relationships

DANIELS FUND ETHICS INITIATIVE – PRINCIPLE 3

- Accountability
  - Accept responsibility for all decisions

DANIELS FUND ETHICS INITIATIVE – PRINCIPLE 4

- Transparency
  - Maintain open and truthful communications

DANIELS FUND ETHICS INITIATIVE – PRINCIPLE 5

- Fairness
  - Engage in fair competition and create equitable and just relationships

DANIELS FUND ETHICS INITIATIVE – PRINCIPLE 6

- Respect
    - Honor the rights, freedoms, views, and property of others

DANIELS FUND ETHICS INITIATIVE – PRINCIPLE 7

- Rule of Law
  - Comply with the spirit and intent of laws and regulations

DANIELS FUND ETHICS INITIATIVE – PRINCIPLE 8

- Viability
  - Create long-term value for all relevant stakeholders

TEAM FORMATION

- Talk to your peers
- Form your semester project teams
- Ideally, size 3

PROJECT IDEATION REVISITED

- Now, you need to agree on one topic in your teams!
- The topic that you choose does not matter that much
- It just has to be somewhat interesting to you and related to this course
- The way you apply the principles and practices you will learn to your project does matter!

QUESTIONS?

See you!