

# Operating System Concepts

Course Code: CSC 2209

Course Title: Operating Systems



**Dept. of Computer Science**  
**Faculty of Science and Technology**

<b>Lecturer No:</b>	<b>01</b>	<b>Week No:</b>	<b>01</b>	<b>Semester:</b>	<b>Fall 2024-25</b>
<b>Assistant Professor</b>	<i>Dr. Rajarshi Roy Chowdhury, rajarshi@aiub.edu</i>				

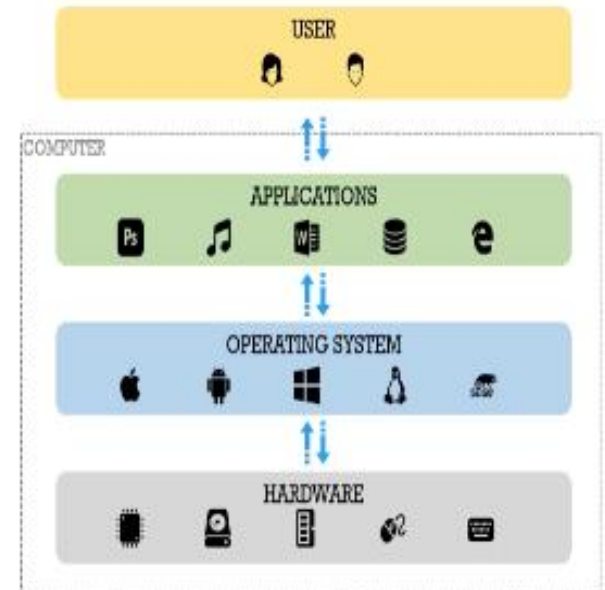
# Lecture Outline



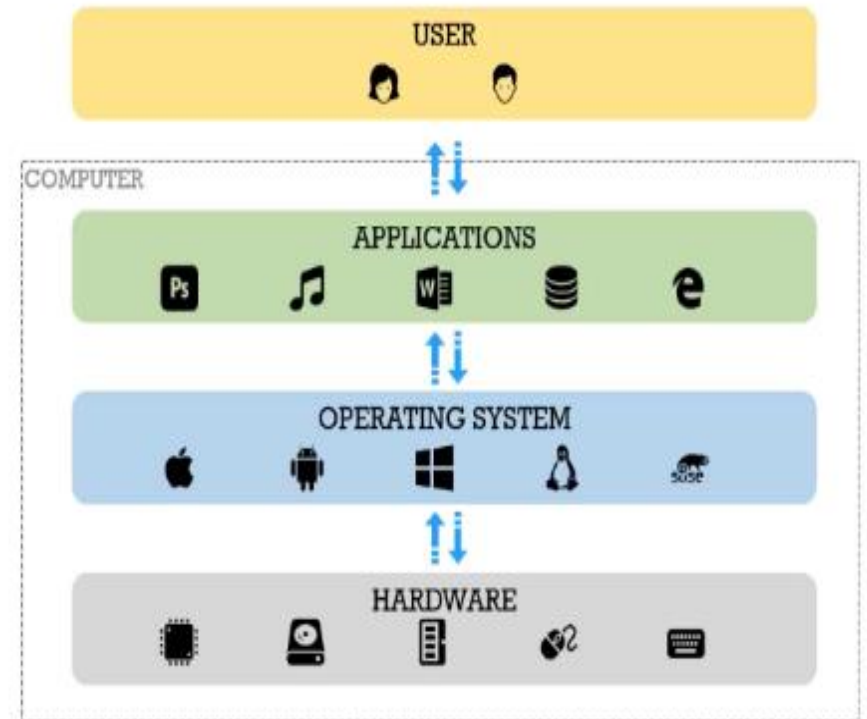
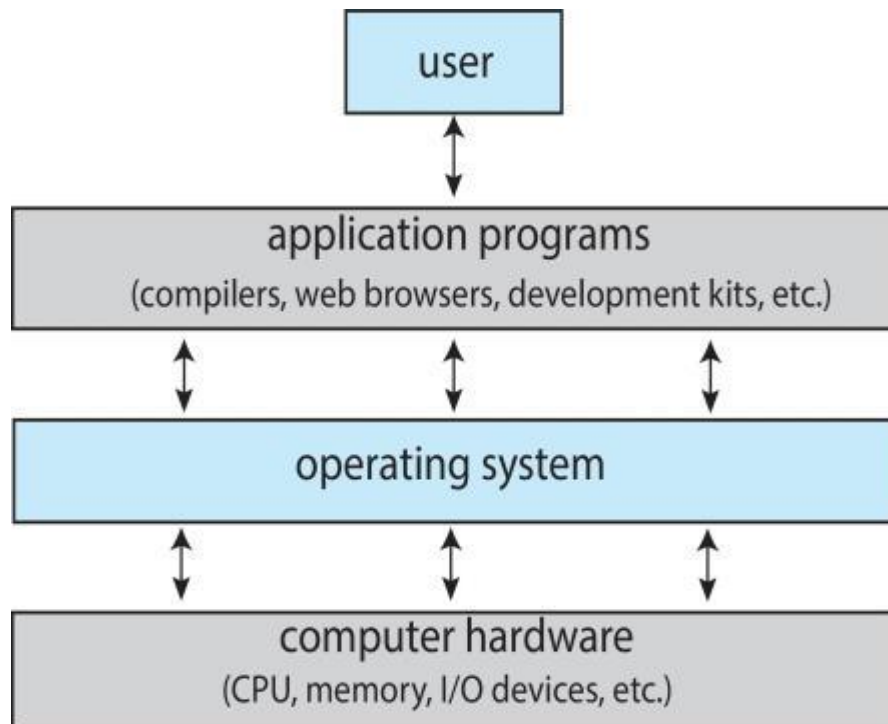
1. Computer System
2. Abstract View of Computer Components
3. What Operating Systems Do
4. Defining Operating Systems
5. Computer System Organization
6. Computer-System Operation
7. Interrupt
8. Computer Startup
9. Interrupt Handling
10. Interrupt-drive I/O Cycle

# Computer System

- ❑ Computer system can be divided into four components:
  - ❑ **Hardware** – provides basic computing resources
    - ❑ CPU, memory, I/O devices
  - ❑ **Operating system**
    - ❑ Controls and coordinates use of hardware among various applications and users
  - ❑ **Application programs** – define the ways in which the system resources are used to solve the computing problems of the users
    - ❑ Word processors, compilers, web browsers, database systems, video games
  - ❑ **Users**
    - ❑ People, machines, other computers



# Abstract View of Computer Components



# What Operating Systems Do

- ❑ Depends on the point of view
- ❑ **Users** want convenience, **ease of use** and **good performance**
  - ❑ Don't care about **resource utilization**
- ❑ But **shared computer** such as **mainframe** or **minicomputer** must keep all users happy
  - ❑ Operating system is a **resource allocator** and **control program** making efficient use of Hardware and managing execution of user programs
- ❑ **Users of dedicate systems** such as **workstations** have dedicated resources but frequently use shared resources from **servers**

# What Operating Systems Do (cont'd)

- ❑ **Mobile devices** like **smartphones** and **tables** are resource poor, optimized for usability and battery life
  - ❑ Mobile user interfaces such as **touch screens, voice recognition**
- ❑ Some **computers have little or no user interface**, such as **embedded computers in devices and automobiles**
  - ❑ Run primarily without user intervention

# Defining Operating Systems

- ❑ Term OS covers many roles
  - ❑ Because of myriad designs and uses of OSs
  - ❑ Present in toasters through ships, spacecraft, game machines, TVs and industrial control systems
  - ❑ Born when fixed use computers for military became more general purpose and needed resource management and program control

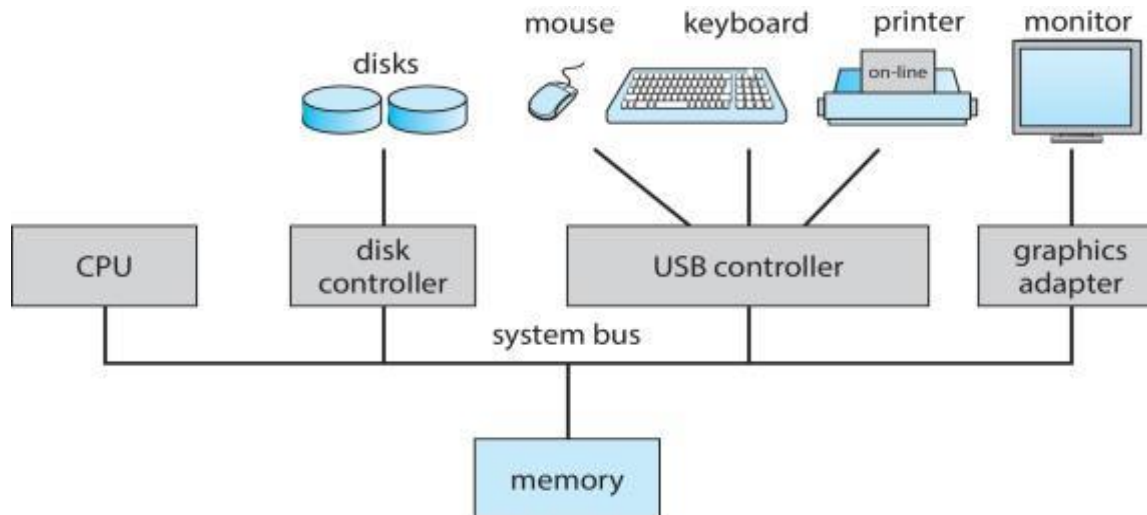
# Operating System Definition (cont'd)

- ❑ No universally accepted definition
- ❑ “Everything a vendor ships when you order an operating system” is a good approximation
  - ❑ But varies wildly
- ❑ “The one program running at all times on the computer” is the **kernel**, part of the operating system
- ❑ Everything else is either
  - ❑ a **system program** (ships with the operating system, but not part of the kernel) , or
  - ❑ an **application program**, all programs not associated with the operating system
- ❑ Today’s OSs for general purpose and mobile computing also include **middleware** – a set of software frameworks that provide addition services to application developers such as databases, multimedia, graphics



# Computer System Organization

- ❑ Computer-system operation
  - ❑ One or more CPUs, device controllers connect through common **bus** providing access to shared memory
  - ❑ Concurrent execution of CPUs and devices competing for memory cycles



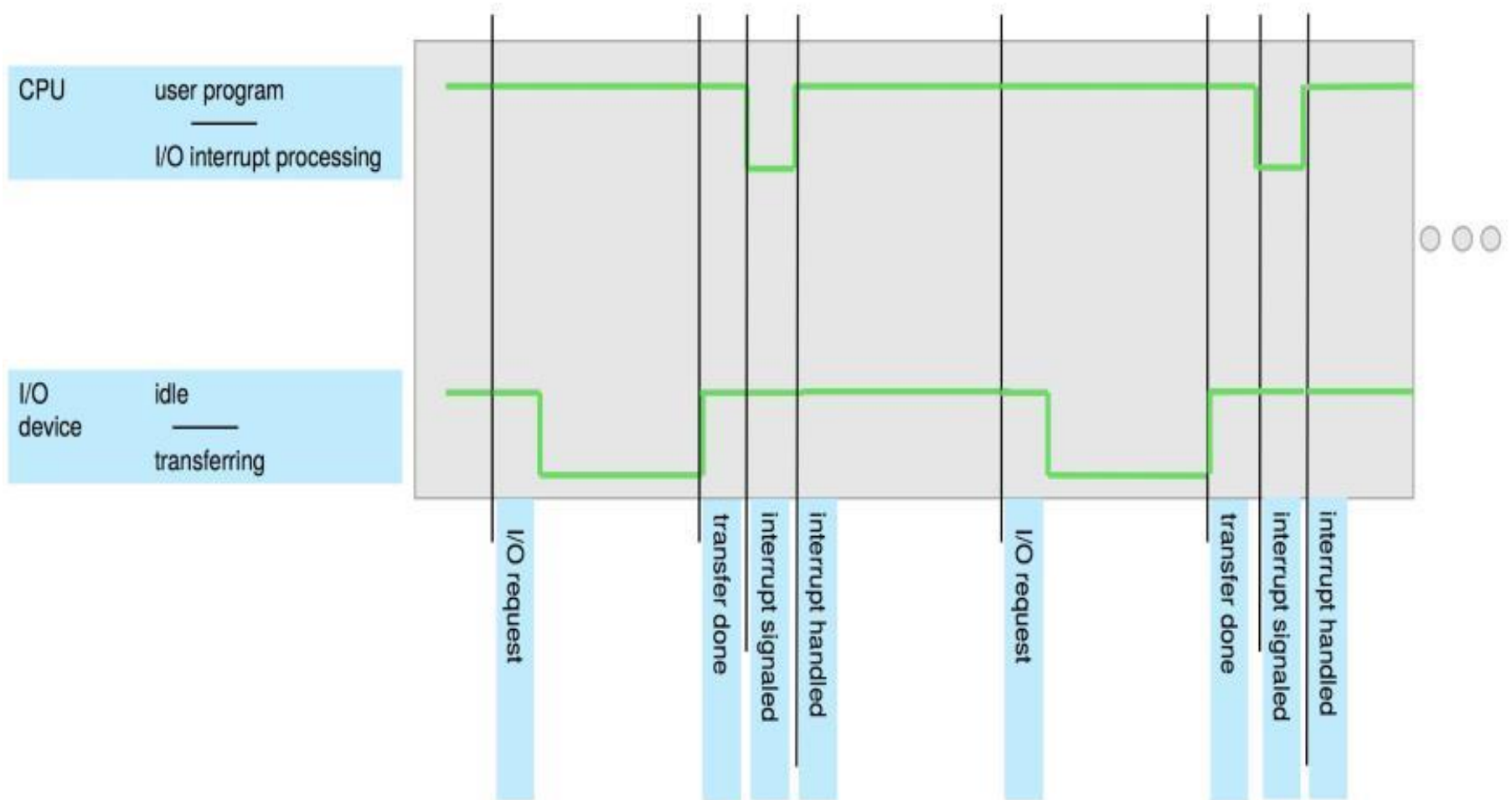
# Computer-System Operation

- ❑ I/O devices and the CPU can execute **concurrently**
- ❑ Each **device controller** is in charge of a particular device type
- ❑ **Each device** controller has a **local buffer**
- ❑ Each device controller type has an operating system **device driver** to manage it
- ❑ CPU moves data from/to main memory to/from local buffers
- ❑ I/O is from the device to **local buffer of controller**
- ❑ Device controller informs CPU that it has finished its operation by causing an **interrupt**

# Common Functions of Interrupts

- ❑ Interrupt transfers control to the interrupt service routine generally, through the **interrupt vector**, which contains the addresses of all the service routines
- ❑ Interrupt architecture must save the address of the interrupted instruction
- ❑ A **trap** or **exception** is a **software-generated interrupt** caused either by an error or a user request
- ❑ An operating system is **interrupt driven**

# Interrupt Timeline



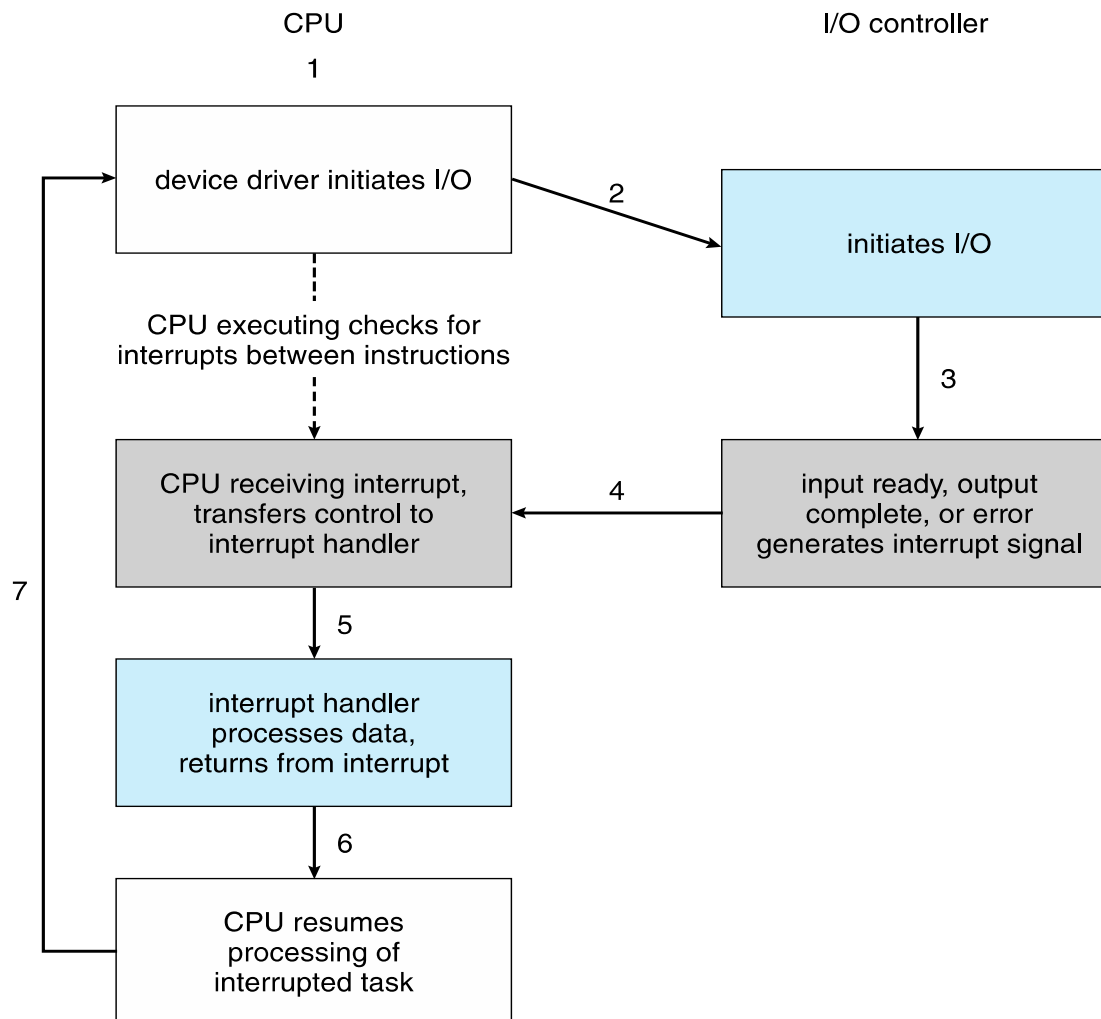
# Computer Startup

- ❑ **bootstrap program** is loaded at power-up or reboot
  - ❑ Typically stored in **ROM** or **EPROM**, generally known as **firmware**
  - ❑ **Initializes all aspects of system**
  - ❑ **Loads operating system kernel and starts execution**

# Interrupt Handling

- ❑ The operating system preserves the state of the CPU by storing registers and the program counter
- ❑ Determines which type of interrupt has occurred:
  - ❑ **Polling** interrupt (CPU keeps polling at regular intervals if a device is ready)
  - ❑ **Vectored** interrupt (I/O device requests for attention)
- ❑ Separate segments of code determine what action should be taken (for each type of interrupt)

# Interrupt-drive I/O Cycle





# Books

- ❑ Operating Systems Concept
  - ❑ Written by Galvin and Silberschatz
  - ❑ Edition: 9<sup>th</sup>





## References

- ❑ Operating Systems Concept
  - ❑ Written by Galvin and Silberschatz
  - ❑ Edition: 9<sup>th</sup>