CSC 4320/6320: Operating Systems

Georgia State University

Course Introduction

Spring 2025

Instructor: Dr. Md Mahfuzur Rahman
Department of Computer Science, GSU

Course Information

Instructor: Md. Mahfuzur Rahman, Ph.D.

Email: mrahman21@gsu.edu

Class Schedule: TR 2:45 PM – 4:30 PM

Location: Classroom South, Room: 103

Instructor Office Hours: MW: 03:45 PM – 05:00 PM, or by appointment.

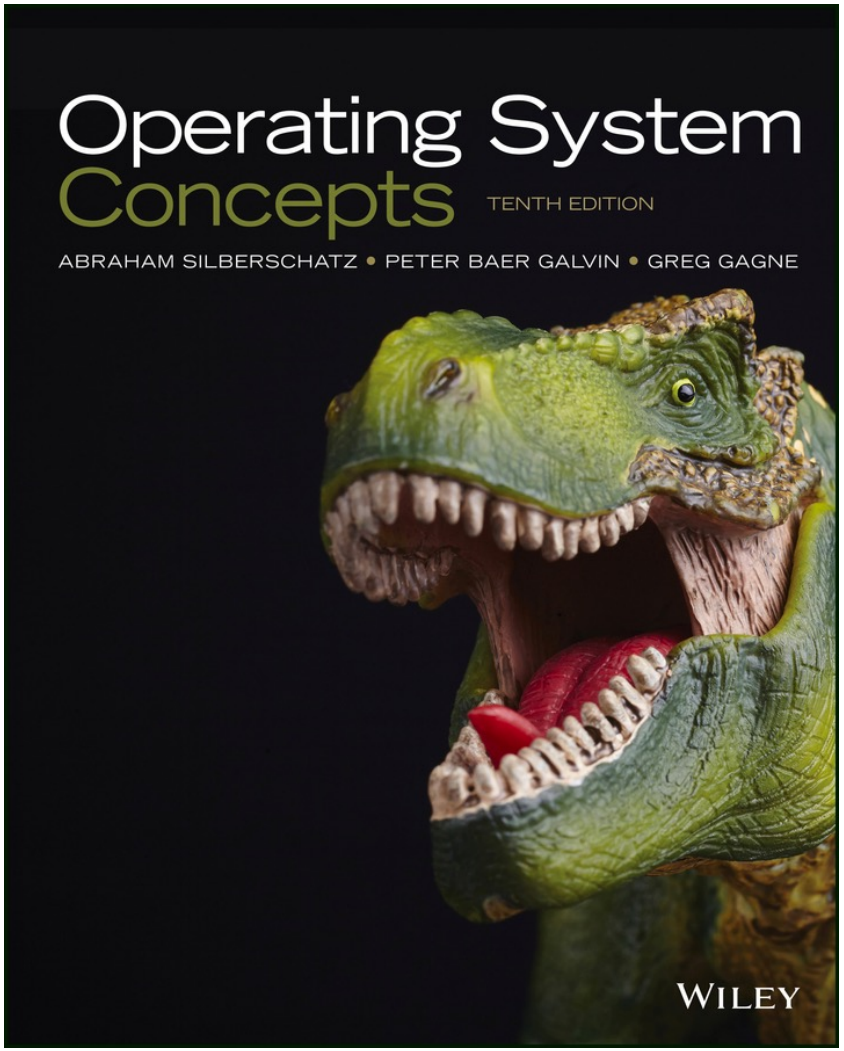
Instructor Office Location: 1 PP, Room: 637

TA: Sri Venkata Naga Sai Kakani

TA Email: skakani1@student.gsu.edu

TA Office Hours: TBA

Textbook



Title: “Operating System Concepts“ 10th Edition by Silberschatz, Galvin, and Gagne

Book Site:
https://os-book.com/OS10/index.html

Prerequisites

- CSC 3320 with grade of C or higher.
- If you do not meet these requirements, please drop.

Grading Policy

- **Homework** (approx. 5): 25% (lowest score will be dropped)
- **In-class Participation:** 10% (in-class quizzes can be given. All quizzes are automatically graded) (lowest score will be dropped)
- **Exam 1 and Exam 2:** 20% each
- **Research Paper Presentation:** 10% (graduate students MS/PhD only)
- **Practice Project:** 15% (graduate students only)
- **Final Exam:** 25% (for undergraduate only)

Course Policy

- Homework must be turned in **iCollege**
- We will use **gcc** compiler and **snowball.cs.gsu.edu** platform for programming parts in assignments.
- For homework, 10% per day will be deducted from your score. Homework will not be graded after a week.
- **Make-Ups:** there will be no make-up examinations for any unexcused absence. No make-up for the final exam.

Course Policy (contd.)

- **Class participation:** In-class (low stakes) pop-quizzes will be taken during the course. (mainly to encourage participation). It will also contribute toward 10% of your grade, as stated earlier
- Any gradable work must be your own.
- No use of Generative AI unless allowed in writing.
- No sharing of course materials on any website
- Read carefully the entire syllabus.

How can you succeed in this course?

- Participate in everything: pretest, posttest, homework, exams, quizzes, etc.
- Listen to the lectures carefully and take notes.
- Study the lecture materials thoroughly.
- Do not memorize things. Grow your knowledge.
- Start doing your homework earlier and submit timely.
- Show your uniqueness in your submission. (No copy, no plagiarism)
- You may be allowed to keep a 1-page paper (cheat sheet) to help you take some notes with you. The exams will be concept-based, will only test your knowledge.

Georgia State University

Let’s start course topics

Disclaimer

The slides to be used in this course have been created, modified, and adapted from multiple sources:

- *The slides are copyright Silberschatz, Galvin and Gagne, 2018. The slides are authorized for personal use, and for use in conjunction with a course for which Operating System Concepts is the prescribed text. Instructors are free to modify the slides to their taste, as long as the modified slides acknowledge the source and the fact that they have been modified.*

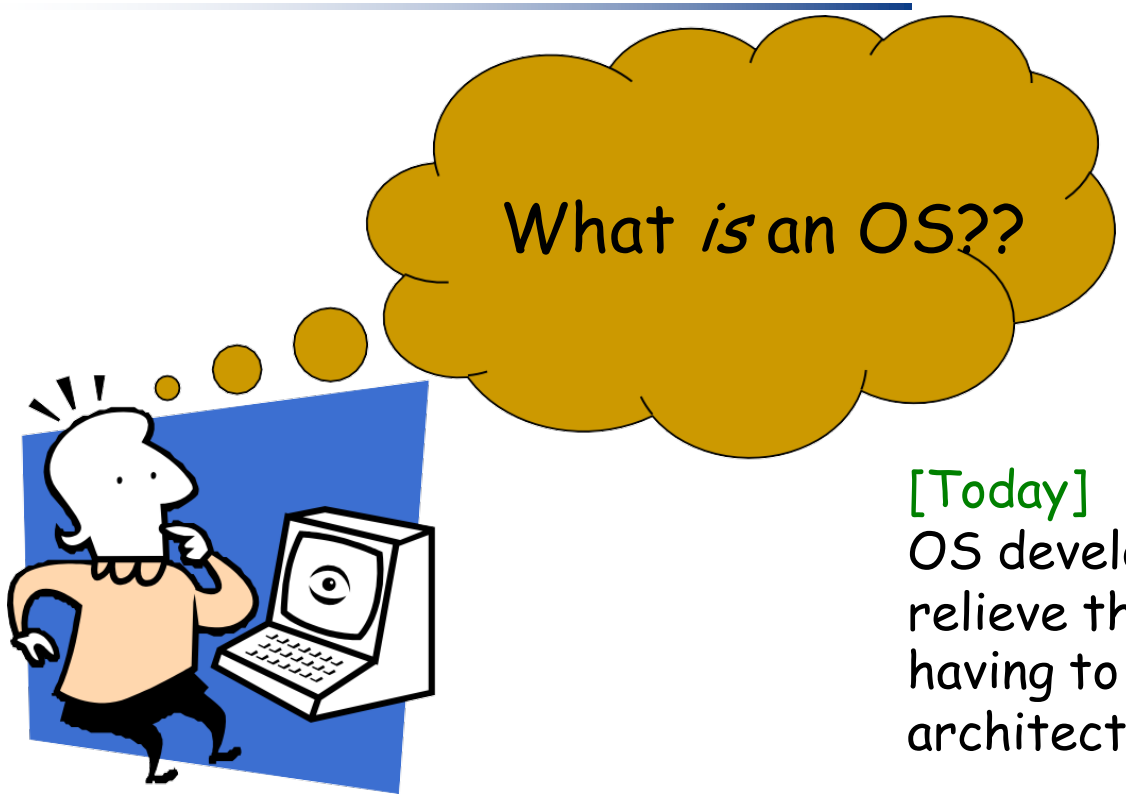
- *Some slides are from Mario Kubek, Assistant Professor, Computer Science, GSU. These slides are marked by MK*.*

What this Course is about…

- The role of the OS in
  - Resource Management
  - Task Management
  - Process Coordination
  - Service Provisioning
- Process Management
  - Process Coordination
  - Synchronization
  - Threads vs. Processes
  - Scheduling
  - Real-time Scheduling
  - Deadlocks

- Memory Management
  - Physical Memory Partition Allocation Virtual Memory
- File Systems
  - Basic FS functions FS Organization FS Implementation
- Virtual Machines
  - Benefits
  - Types and implementations
  - Virtualization

Introduction



[Historically]
OS development was driven by the need to efficiently use very expensive hardware.

[Today]
OS development is driven by the goal to relieve the users and programmers from having to deal with the details of the architecture.

Hard to define precisely – it evolved with the need of people to use computers.

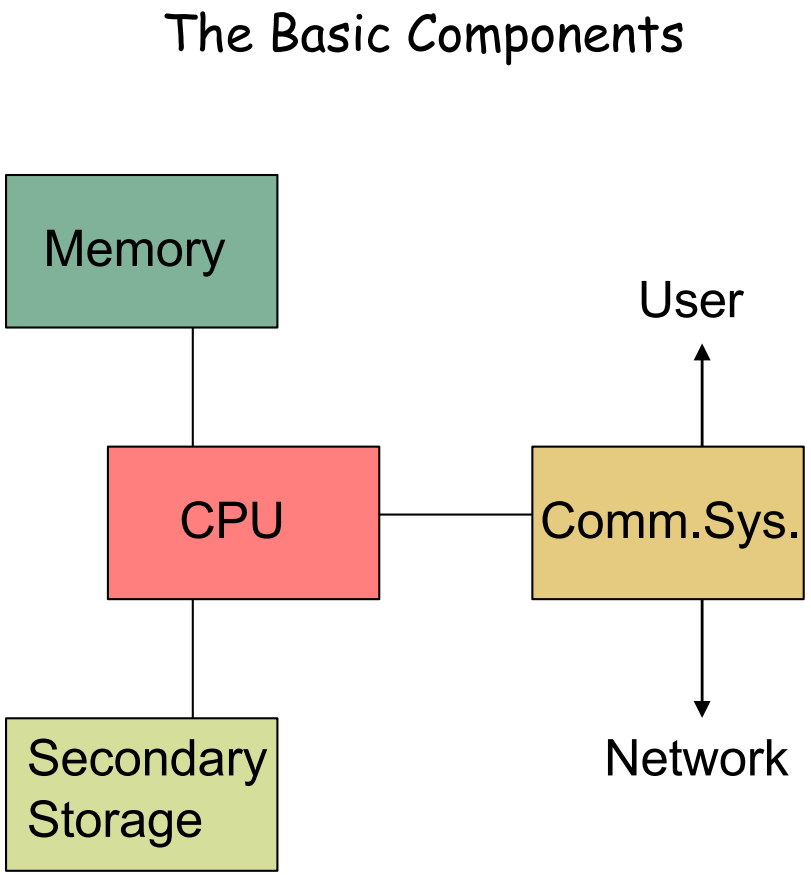
Why do we study OS?

Georgia State University

- Almost all code runs on top of an operating system
- Knowledge of how operating systems work is crucial to proper, efficient, effective, and secure programming.
- Understanding how operating systems (OSs) control computer hardware and what features they offer to applications is crucial. It benefits:
  - People who develop operating systems
  - those who create software that runs on these operating systems or use those applications.

Bridging between HW and Apps

- Most Systems are based on the John von Neumann architecture principles, i.e., the *stored program computer*.
  - CPU
  - Main Memory
  - Communication Subsystem
  - Secondary Storage (i.e., Disk)

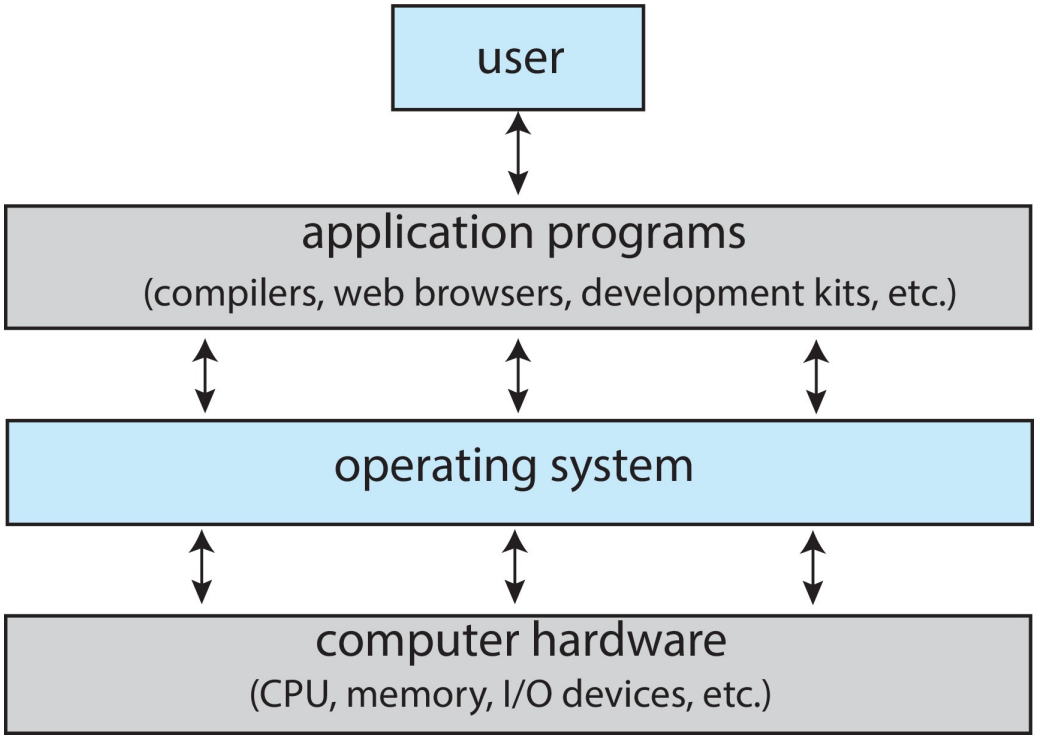


Computer System Structure

- 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

What is an OS?

An operating system is “fill in the blanks” between … and …?

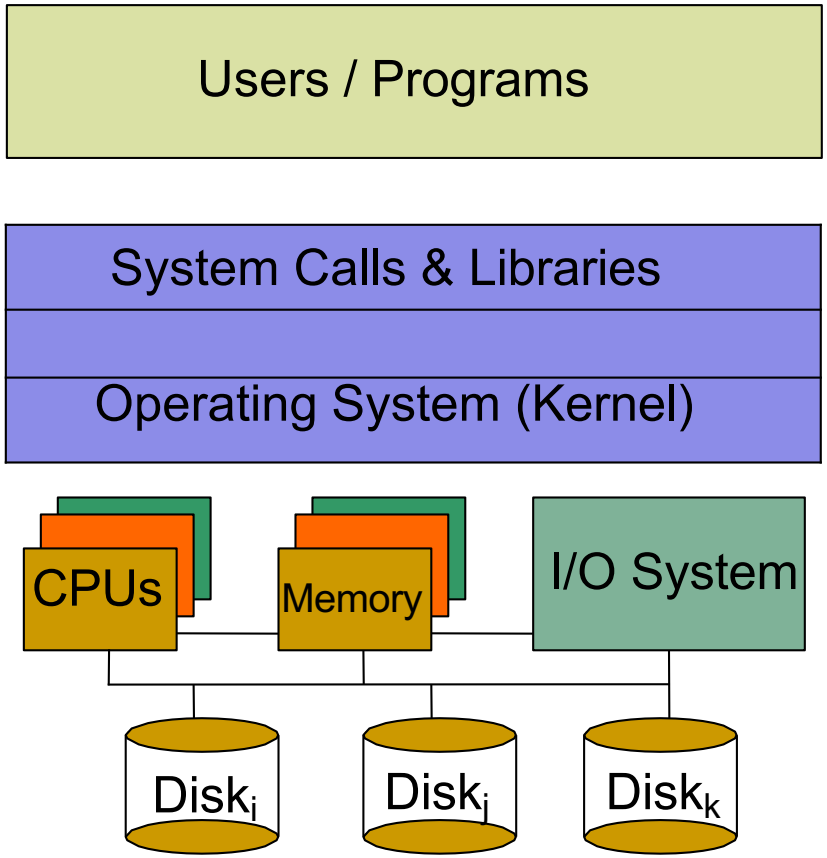


Filling the Gaps

- The OS must fill the gaps between what is provided by the HW and what is required by the Applications, i.e.:
  - *Unify view of memory*
  - *Transfer data between CPU and different devices*
  - *Keep track of file locations*
  - *Provide HW / Platform Independence*
  - *Mediate Resource Usage*

The OS relieves programmers of having to deal with the specifics of each device and the system architecture. □ Abstraction

The Principle of Abstraction:



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 HW and managing execution of user programs
- Users of dedicated systems such as **workstations** have dedicated resources but frequently use shared resources from **servers**
- 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 home devices and automobiles
  - Run primarily without user intervention

Operating System Definition

- 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 OSes for general purpose and mobile computing also include ***middleware*** – a set of software frameworks that provide additional services to application developers such as databases, multimedia, graphics
  - Apple's IOS and Google's Android

Let's recall together

1. The operating system kernel consists of all system and application programs in a computer.

A. True
B. False

2. All computer systems have some sort of user interaction.

A. True
B. False

Think about embedded computers

Questions

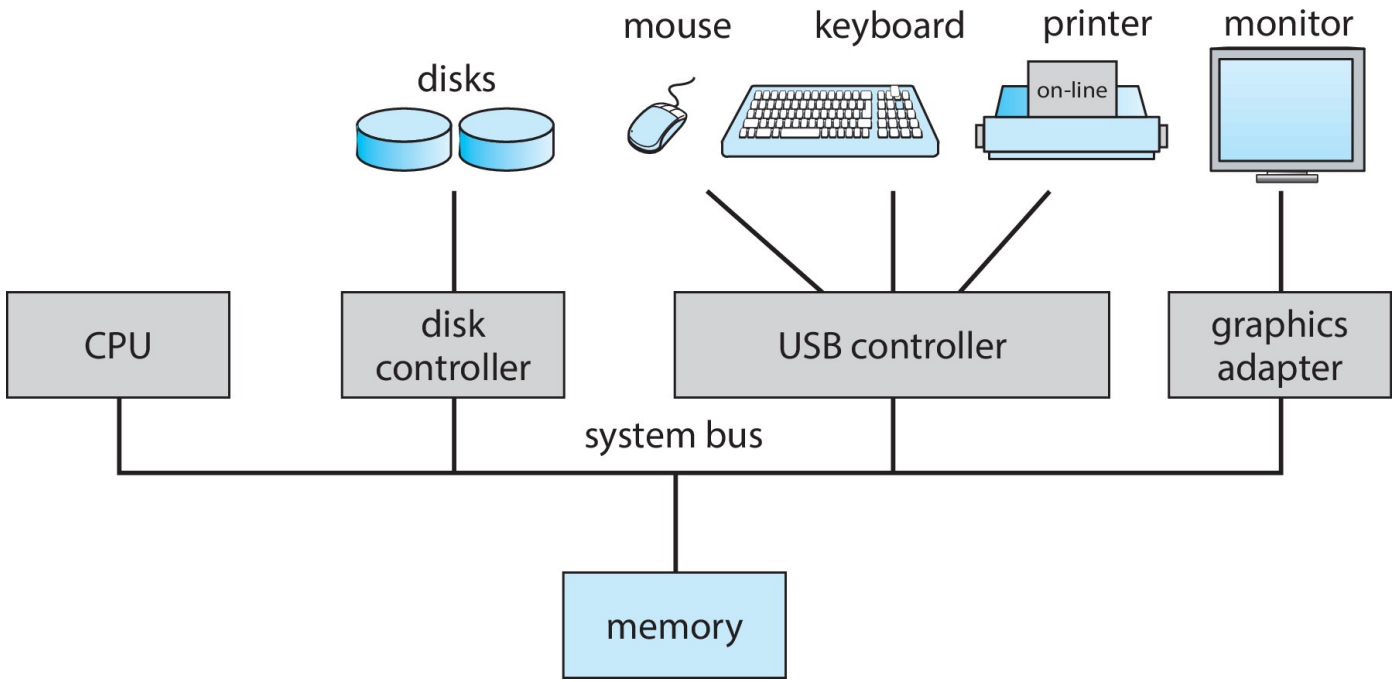
1. _____ is a set of software frameworks that provide additional services to application developers.

A) System programs
B) Virtualization
C) Cloud computing
D) Middleware

Ans: Middleware

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



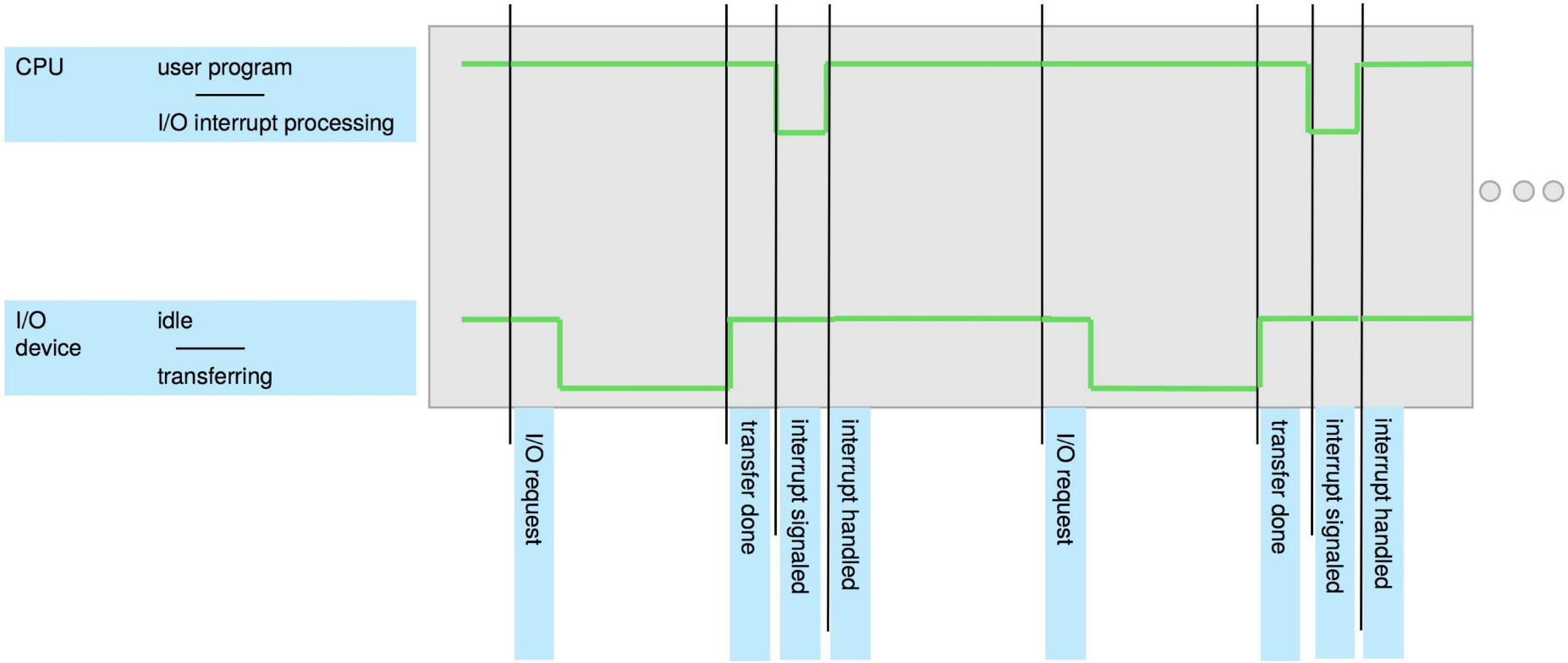
Computer-System Operation

- 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



Interrupt Handling

- The operating system preserves the state of the CPU by storing the registers and the program counter
- Determines which type of interrupt has occurred:
- Separate segments of code determine what action should be taken for each type of interrupt

Georgia State University

Questions?