Operating System & Graphical user Interface

Lecture 2

AL-Zahraa College of Medicine

University of Basrah

Objectives

- Define an operating system and explain its essential role
- Describe key OS resource management functions (CPU, memory, devices, storage, processes)
- Identify and compare common desktop, mobile, and server operating systems
- Understand factors in choosing an operating system
- Differentiate between GUI and CLI interfaces
- Demonstrate proper mouse techniques for GUI interaction

Operating System

- Every computer, from smartphones to supercomputers, has an operating system. Even game consoles, cars, and some appliances include an OS.
- An **operating system** is the main software that manages all hardware and software on a computer. It acts as a bridge between the user, applications, and the hardware.
- Without it, a computer cannot function because hardware does not understand human commands directly.
- The **OS** ensures that every program gets the resources it needs such as CPU time, memory space, and access to storage without crashing into other program.

Importance of the Operating System in Managing Hardware and Software Resources

❑ Resource Management

- Resource management is a core function of an operating system (OS). It ensures that hardware and software resources are allocated efficiently, fairly, and safely among all running programs.

❑ Key Roles of the OS in Resource Management

A. CPU (Processor) Management

- **Scheduling:** Determines which process runs at any given time.
- **Multitasking:** Allows multiple applications to run simultaneously without conflict.
- **Example:** When you are watching a YouTube video, downloading a file, and typing in Word at the same time, the operating system rapidly switches the CPU between these tasks. It schedules tiny time slices for each process so everything appears to run simultaneously this is how multitasking works.

Importance of the Operating System in Managing Hardware and Software Resources

B. Memory Management

- Assigns memory to programs and prevents conflicts between them.
- Uses virtual memory when physical RAM is insufficient.
- Provides caching for faster access to frequently used data.
- **Example:** Opening multiple applications in Windows, the OS allocates memory efficiently to prevent crashes.

- *Caching means:* ***Storing frequently used data in a fast-access location***

so the system can retrieve it quickly without recalculating or reloading it from slower storage.

Importance of the Operating System in Managing Hardware and Software Resources

C. Device Management

- Controls input/output devices such as keyboards, mice, displays, printers, and network cards.
- Uses **device drivers (***Refer to slide 9 for details***),**to communicate between software and hardware.
- Ensures multiple programs can safely and efficiently share devices.
- **Example:** Managing a printer queue while coordinating CPU and memory usage.

Importance of the Operating System in Managing Hardware and Software Resources

D. Storage and File System Management

- **File Organization:** Stores data in files and directories for easy access.
- **Access Control:** Sets permissions to protect files from unauthorized access.
- **Storage Allocation:** Tracks used and free space to prevent data corruption.
- **File Operations:** Supports creating, reading, writing, copying, moving, and deleting files.

❑ **Examples:** When you save a document in Windows, the OS stores it in a specific folder on your hard drive, keeps track of its location, and ensures that only authorized users can access or modify it. Later, when you open the document, the OS retrieves it quickly from the storage.

E. Process Management

- A **process** is a program in execution. The OS is responsible for creating, scheduling, and terminating processes.

❑ Key Functions:

1. **Process Creation and Termination:** Starts programs and cleans up after they finish.
2. **Process Scheduling:** Decides which process gets CPU time based on priority.
3. **Process Synchronization:** Ensures processes that share resources don't conflict.

❑ Example:

- When you open a web browser, the OS creates a process for it, allocates CPU and memory, and manages its execution. If you open multiple programs, the OS schedules CPU time for each, allowing all of them to run simultaneously. When you close the browser, the OS terminates its process and frees the resources.

Device Drivers

A **device driver** is a small software program that allows the operating system to communicate with hardware devices.

❑ Why They Are Important

- The OS cannot directly control hardware, it needs drivers to translate commands.
- Without the correct driver, the hardware will not function properly or may not work at all.

Examples

- A **printer driver** allows Windows or macOS to send documents to a specific printer model.
- A **Wi-Fi driver** enables the computer to connect to wireless networks.

Basics of Common Operating Systems

- Operating systems such as **Microsoft Windows**, **Apple macOS**, **Linux distributions**, **and** **Chrome OS** are the dominant platforms in personal computing.

Types of Operating Systems

- Operating systems can be classified based on their use and device type.

A. Desktop and Laptop OS

1. Microsoft Windows

Market Share: The dominant leader in the global desktop market.

❑ Key Characteristics:

- User-Friendly GUI: Known for its familiar graphical user interface (Start Menu, Taskbar).
- Wide Software Compatibility: The largest library of commercial software, especially for gaming and business applications.
- Hardware Variety: Runs on a vast array of hardware from numerous manufacturers (Dell, HP, Lenovo, etc.).
- Common Versions: Windows 10, Windows 11.

2. macOS

Market Share: The second most popular desktop OS.

❑ Key Characteristics:

- Exclusive to Apple Hardware: Only runs on Apple's Mac computers (iMac, MacBook, Mac Pro).
- User Experience: Renowned for its sleek, intuitive, and polished user interface.
- Ecosystem Integration: Seamlessly integrates with other Apple devices like iPhone, iPad, and Apple Watch.
- Stability & Security: Generally considered less vulnerable to malware than Windows.
- Common Versions: macOS Sonoma, macOS Ventura.

3. Linux

❑ Key Characteristics:

- **Open-Source:** The core code is free to use, modify, and distribute.
- **Distributions:** Comes in many different "flavors" or distributions, each with its own focus (e.g., user-friendliness, performance, server use).
- **Customizability & Control:** Highly customizable and favored by developers, system administrators, and tech enthusiasts.

4. ChromeOS

- Google Chrome OS is a web-based operating system. With Chrome OS, Applications are primarily web apps, files are stored in the cloud (like Google Drive).
- Hardware Specific: It is designed specifically for a class of devices called Chromebooks, which are produced by partners like Acer, ASUS, Dell, Lenovo, and Google itself (Pixelbook).

❑ **OS vs. Application:** This is a critical distinction. Chrome OS is the entire operating system that powers the computer. The Chrome Browser is a single application that can be installed on other operating systems like Windows, macOS, and Linux.

B. Mobile Operating Systems

These power smartphones and tablets.

1. Android

- **Market Share:** The most popular mobile OS.

❑ Key Characteristics:

- **Open-Source:** The Android Open Source Project allows manufacturers to customize the OS.
- **Hardware Diversity:** Runs on devices from many companies (Samsung, Google, OnePlus, Xiaomi, etc.).
- **Google Ecosystem:** Deeply integrated with Google services (Gmail, Maps, Play Store).
- **Customizability:** Offers a high degree of customization for users.

2. iOS

- **Market Share:** The second major mobile OS.

❑ Key Characteristics:

- **Exclusive to Apple Hardware:** Only runs on iPhones and iPads.
- **Walled Garden:** A tightly controlled App Store ensures app quality and security, but limits user freedom compared to Android.
- **User Experience:** Known for its smooth performance, intuitive interface, and cohesive design.
- **Ecosystem Integration:** Works flawlessly with other Apple devices and services (Mac, Apple Watch, AirDrop).

Basics of Common Operating Systems

C. Server Operating Systems

- These run on powerful computers that provide services (websites, databases, files) to other computers over a network. Examples: Windows Server and Linux Server.
- Manages large networks, databases, and web services.

D. Other / Embedded Operating Systems

- These are lightweight operating systems designed for specific devices.
- Real-Time Operating Systems (RTOS): Used in time-critical systems like car airbags, medical devices, and industrial controllers.

Choosing Operating Systems

Running Multiple Operating Systems

- You can run or test programs on more than one operating system. You might want to do this to:
  - Test software across different platforms
  - Use an application that is available only on a specific OS
  - Access development tools that perform differently on different systems

- Windows and Linux can run on most modern hardware.
- Although macOS and iOS run only on Apple devices, Apple hardware can still support Windows and Linux through virtualization or dual-boot configurations.
- Ultimately, selecting an operating system often comes down to price, hardware compatibility, and personal preference.

The User Interface (UI)

- The operating system provides the **user interface (UI)** you use to interact with your device.
- Early computers used **command-driven interfaces** like MS-DOS, where users had to type exact commands. This was powerful but difficult for most people.
- Later, **menu-driven interfaces** appeared, allowing users to choose commands from on-screen menus instead of typing them. These were easier but still not very intuitive.
- Modern operating systems such as **Windows** and **macOS** use a **Graphical User Interface (GUI)**, which displays icons, windows, and buttons and uses point-and-click actions, making computers far more user-friendly.

The User Interface (UI)

- The User Interface (UI) is the component of an OS that allows users to interact with the computer and its programs. Without a UI, users would not be able to give commands or access applications easily.

- There are two primary types of user interfaces that users encounter: the Graphical User Interface (GUI) and the Command Line Interface (CLI). Each offers a different way to control and manage a computer's functions.

The User Interface (UI)

A. Graphical User Interface (GUI): A **GUI** is a visual way of interacting with a computer. Instead of typing commands, you use icons, buttons, menus, and windows to control the system..

- **Interaction:** Use a **mouse**, **touchpad**, **or finger** to point, click, drag, and drop.
- **Learning Curve:** Easy to learn because visual cues like a trash can icon representing deletion help users understand functions quickly.
- **Experience:** A GUI makes using the computer feel visual and easy. When you click something, you instantly see feedback such as animations, progress bars, or windows opening and closing. Your files and apps are displayed as icons and folders, so you can find and manage them quickly.
- **Downside:** Can be slow for complex tasks and **resource-heavy**, requiring significant processing power for graphics.

The User Interface (UI)

B. The Command-Line Interface (CLI): A **CLI** lets you interact with the OS by typing text commands instead of using visuals.

- **Interaction:** Type commands using a **keyboard** (e.g., mkdir, rm, ipconfig).
- **Learning Curve:** Steeper than GUI; you must remember commands and syntax.
- **Experience:** Very **powerful and fast** for advanced tasks, automation, and system administration.
- **Downside:** Not visually intuitive; errors can happen easily if a command is mistyped. It can be intimidating for beginners.

Which is Better?

Neither is universally "better." They serve different purposes and often complement each other.

- **Use a GUI** for everyday tasks: browsing the web, writing documents, managing personal files. It's user-friendly and visually intuitive.
- **Use a CLI** for system administration, software development, automating tasks, or managing remote servers. It's powerful, precise, and efficient.

❑ Most modern operating systems provide both. For instance, **Windows:** Users work with the desktop GUI, while advanced tasks can be done via PowerShell or Command Prompt.

Using Mouse Techniques in an Operating System

The mouse is a primary input device in modern operating systems. It allows users to interact with graphical elements in a GUI (Graphical User Interface). Learning proper mouse techniques is essential for efficient navigation and task execution.

1. Basic Mouse Components

- Left Button: Usually selects items, opens files, or clicks buttons.
- Right Button: Opens context menus with additional options.
- Scroll Wheel: Scrolls vertically (and sometimes horizontally) through documents or web pages.

Using Mouse Techniques in an Operating System

2. Basic Mouse Actions

A. Pointing

- Move the mouse to **position the pointer** on a specific icon, button, or menu item.
- **Example:** Pointing to a file on the desktop before opening it.

B. Clicking

1. **Single Click (Left Button):**
   a) Selects a file, folder, or menu item.
   b) Example: Click once on a folder to highlight it.
2. **Double Click (Left Button):**
   a) Opens a file, folder, or application.
   b) Example: Double-click a Word document to open it.
3. **Right Click:**
   a) Opens a **context menu** with options related to the selected item.
   b) Example: Right-click a file → "Copy," "Rename," or "Delete."

Using Mouse Techniques in an Operating System

C. Dragging and Dropping

- **Drag:** Click and hold the left button while moving the pointer.
- **Drop:** Release the button to place the item in a new location.
- **Example:** Moving a file from one folder to another

D. Scrolling

- Use the **scroll wheel** to move up or down a page.
- In some mice, tilting the wheel scrolls horizontally.
- **Example:** Browsing a long webpage or document.

E. Selecting Multiple Items

- **Click + Drag:** Draw a box around multiple icons to select them.
- **Shift + Click:** Select a range of files in a folder.
- **Ctrl + Click:** Select multiple non-adjacent files individually.

Using Mouse Techniques in an Operating System

F. Other Techniques

- **Hovering:** Pointing at an item without clicking shows tooltips or previews.
- **Context Menus:** Right-click for shortcuts to common actions.
- **Double-click vs Single-click:** Different systems may allow single-click to open items; knowing the system convention is important.

❑ Example Tasks Using Mouse Techniques

- Open a folder: **Double-click**
- Move a file: **Click + drag**
- Rename a file: **Right-click → Rename**
- Select multiple files: **Shift + Click or Ctrl + Click**
- Scroll a webpage: **Scroll wheel**

Summary

- The lecture explained the role of operating systems as the essential software that manages computer hardware and software resources.
- It discussed how the OS handles CPU scheduling, memory allocation, device coordination, storage management, and running processes to keep the system efficient and stable.
- Major operating systems were compared in terms of features and market presence.
- The session also clarified the differences between GUI and CLI interfaces and introduced basic mouse techniques to support effective user interaction with a computer.

End-of-Session Questions

1. What is an operating system and why is it essential for computer operation?

2. Why might a user choose a command-line interface over a graphical user interface for certain tasks?

3. What factors should a medical student consider when choosing between Windows, macOS, or ChromeOS for their academic needs?

4. Describe three different mouse techniques and explain when each would be used in typical computer operation.

Textbooks

1. Evans, A., Martin, K., & Poatsy, M. A. (2023). *Technology in Action* (17th ed.). Pearson.

2. Silberschatz, A., Galvin, P. B., & Gagne, G. (2024). *Operating System Concepts* (11th ed.). Wiley.