

# Operating System Concept

## Introduction

Lecturer: Mashal Rokhan

Master of Science in Computer Science

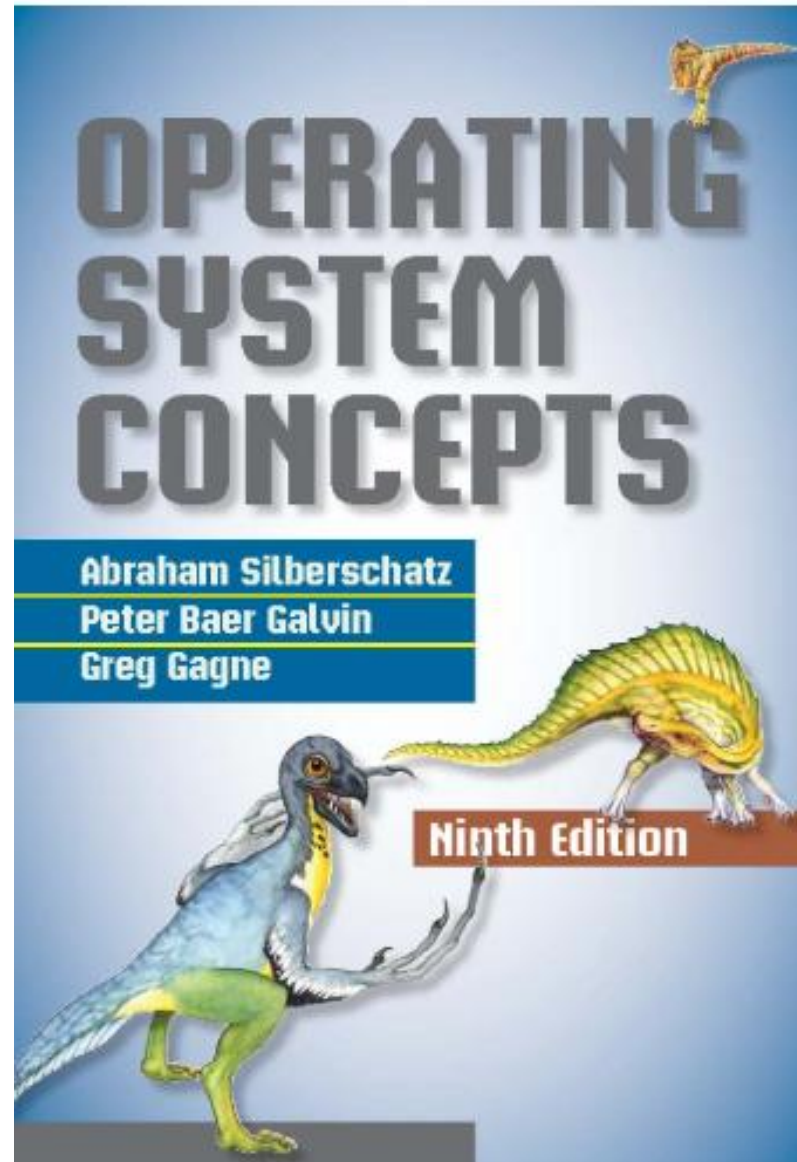
Email: [m.rokhan@kardan.edu.af](mailto:m.rokhan@kardan.edu.af) / [mashalrokhan@gmail.com](mailto:mashalrokhan@gmail.com)

# Grading

- Midterm: 25%
- Final: 50%
- Projects/Assignments: 10%
- Homework and Quizzes: 10%
- Attendance Marks (LMS System attendance will auto calculate your marks)
- If you don't fulfill 75% attendance than system will count you not eligible for exams)

|                 |   |       |
|-----------------|---|-------|
| Full Attendance | 5 | Marks |
| 1 day short     | 4 | Marks |
| 2 day short     | 3 | Marks |
| 3 day short     | 2 | Marks |

# Reference Text Book



# Focus of the Course

- This is a **Concept** based course of Operating System.
- Subject overview and its Domain.
- Software Categories
- System Software
- Operating System
- Utility Program
- Device Driver
- Application Software
- Operating System Foundation Concepts

# What is Operating System (OS)?

- An **Operating System (OS)** is the **foundational software** that **enables hardware components** to **interact** with **software applications** and provides a **user interface** for **communication**.
- It **MANAGES** computer **hardware, software resources**, and provides **special services** for computer programs.

# Key Points to Cover:

- **Definition:** An operating system is a **collection of software** that **manages computer hardware** and provides services for computer programs.
- **Role:** The OS serves as an **intermediary** between the **hardware** and the **applications** running on a computer.
- It controls the execution of programs, manages resources such as **memory, CPU time, and input/output devices**.
- **Examples of Operating Systems:** Windows, macOS, Linux, Android, iOS.
- **Importance:** Without an operating system, **users and applications would not be able to interact with the hardware effectively**. The OS ensures that resources are allocated efficiently, making multitasking and complex operations feasible.

# Domain of Operating System

- **Definition:** The domain of Operating Systems focuses on **understanding** how they work, how they are designed, and how they function.
- **Key Areas:**
  - Working principles (processes, memory, scheduling)
  - System design and architecture
  - Smooth and efficient device operation

# Key Areas in the OS Domain:

## **Process Management:**

The OS is responsible for managing processes, which are programs in execution. It decides the allocation of CPU time for each process, multitasking, and process synchronization.

## **Memory Management:**

OS handles the allocation and deallocation of memory for programs and ensures memory protection.

It also involves virtual memory techniques for managing large programs.

## **File System Management:**

Operating systems handle the storage, retrieval, and organization of data.

## **Device Management:**

- The OS manages hardware devices such as printers, monitors, disk drives, and input devices.



# Key Areas in the OS Domain:

## **Security and Access Control:**

Security is a core function of the OS. It manages user authentication, permissions, and encryption to protect system data from unauthorized access.

## **Networking:**

OS supports network communication, providing protocols like TCP/IP to allow computers to communicate over a network (local or internet).

## **User Interface:**

The OS provides a user interface to interact with the system. This could be a command-line interface (CLI) or a graphical user interface (GUI).

## **Scheduling:**

Operating systems handle the execution of multiple processes concurrently, ensuring efficient CPU usage. Scheduling algorithms are used to decide the execution order.

# Core Concepts in the OS Domain

## **Kernel:**

The core part of the OS that manages resources, memory, and processes.

## **User Space vs. Kernel Space:**

User space is where application programs run, while kernel space is protected and used for system-level functions.

## **System Calls:**

The interface between the application and the OS, allowing programs to request services like input/output operations, memory allocation, etc

# Software Categories

Software can be broadly classified into several categories, each serving different purposes and functions. The most common categories are:

# Software Categories

## 1. System Software:

System software is designed to manage hardware and provide a platform for running application software. It serves as an intermediary between the hardware and application software.

### Examples:

- 1 - Operating Systems (OS):** Windows, macOS, Linux, etc.
- 2 - Device Drivers:** Software that allows the OS to communicate with hardware components like printers, graphics cards, etc.
- 3 - Utility Programs:** Tools that perform maintenance tasks (e.g., disk management, antivirus software).

### Functions:

- Manages computer resources (CPU, memory, storage, etc.)
- Provides a user interface (command line or graphical interface).
- Controls hardware components and peripheral devices.



# Software Categories

## 2. Application Software

Application software is designed to perform specific tasks or solve particular problems for the user. It interacts with system software to perform user-driven functions.

### Examples:

- **Productivity Software:** Microsoft Office (Word, Excel, PowerPoint), Google Docs.
- **Multimedia Software:** Adobe Photoshop, VLC Media Player.
- **Web Browsers:** Google Chrome, Mozilla Firefox, Safari.
- **Games and Entertainment:** Video games, movie players



### Functions:

- Performs tasks such as word processing, database management, web browsing, gaming, etc.
- Provides functionality as per user needs (e.g., word processing, graphic design, etc.).

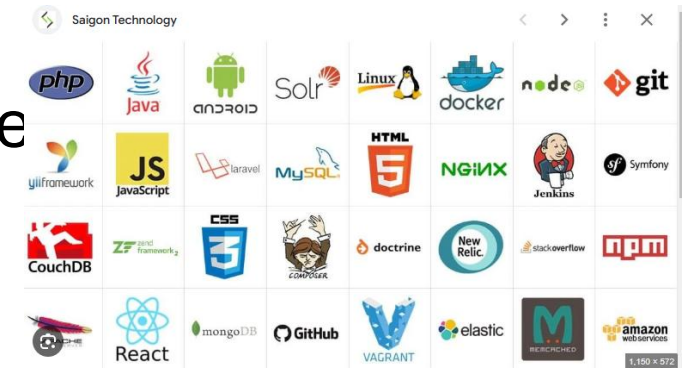
# Software Categories

## 3. Development Software (Development Tools)

Development software, or tools, is used by software developers to create, test, and maintain other software applications.

### Examples:

- **Integrated Development Environments (IDEs):** Visual Studio, Eclipse, Notebook++ etc.
- **Compilers and Interpreters:** GCC, Python Interpreter



# Software Categories

## 4. Middleware

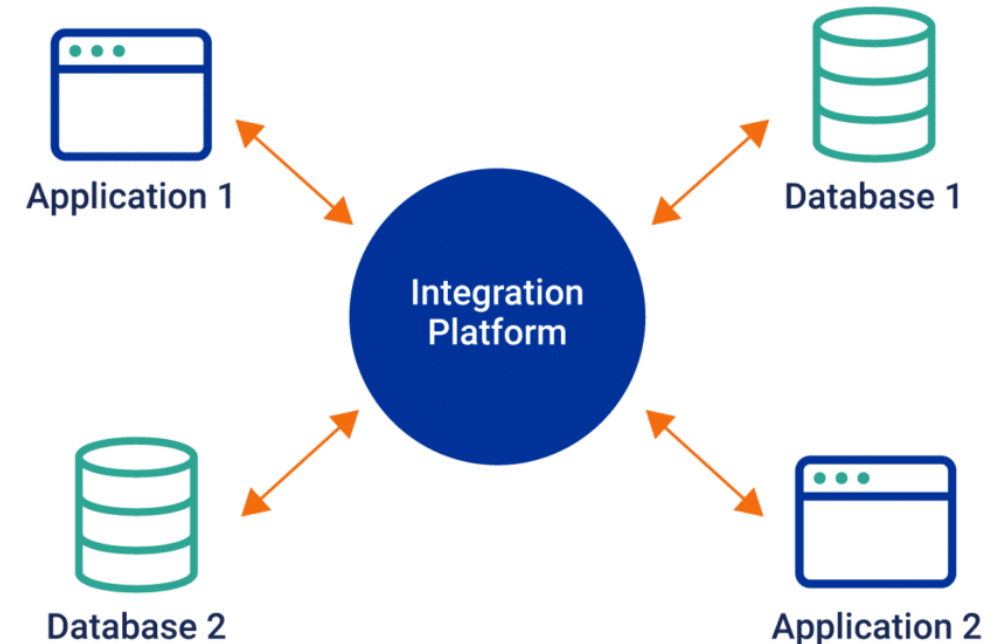
Middleware is software that connects different software applications or services, enabling them to work together seamlessly. It acts as a bridge between application software and system software.

### Examples:

- **Database Middleware:** Oracle Middleware,
- **Web Servers:** Apache HTTP

### Functions:

- Facilitates communication between different systems, databases, and applications.
- Supports distributed computing by enabling integration between different software and hardware systems.



# Software Categories

## 5. Embedded Software

Embedded software is a specialized category of software designed to run on embedded systems — hardware with dedicated functions, often with limited computing resources.

### Examples:

- **Firmware:** Software built into devices like microcontrollers, sensors, and consumer electronics (e.g., **TVs, washing machines**).
- **IoT Devices:** Software running on Internet of Things devices like **smart thermostats, wearable fitness trackers**.

### Functions:

- Controls hardware directly and is tightly integrated with the device's functionality.
- Often runs in real-time or near real-time environments, ensuring quick response times.
- Typically optimized for limited resources such as memory and processing power.





# Software Categories

## 6. Network Software

Network software allows computers and devices to communicate over a network. It includes both software for network communication and software used to manage and configure networks.

### Examples:

- **Networking Protocols:** TCP/IP, HTTP, FTP, DNS.
- **Network Management Tools:** Wireshark, SolarWinds.
- **Firewall Software:** ZoneAlarm, pfSense.

### Functions:

- Enables data transfer and communication between devices over local area networks (LAN), wide area networks (WAN), or the internet.
- Provides services like routing, packet switching, security (firewalls, VPNs), and network monitoring.

# Software Categories

## 7. Business Software

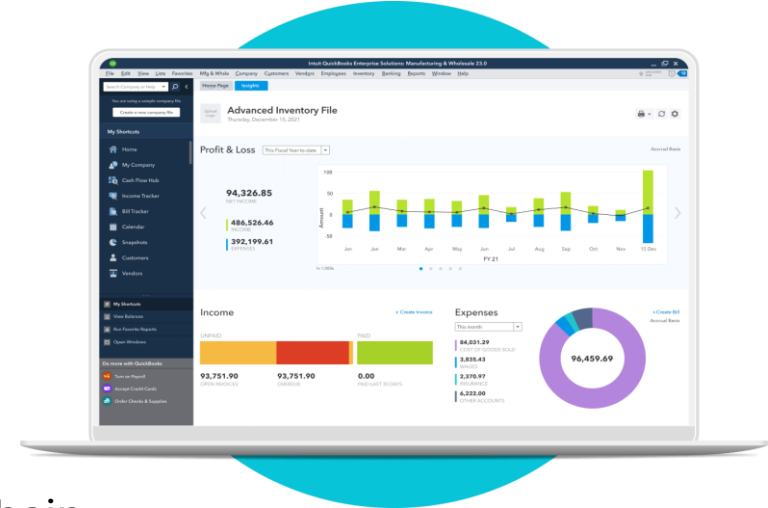
Business software refers to tools designed to help organizations manage various business processes, from accounting to human resources management.

### Examples:

- **Enterprise Resource Planning (ERP):** SAP, Oracle ERP.
- **Customer Relationship Management (CRM):** Salesforce, HubSpot.
- **Accounting Software:** QuickBooks

### Functions:

- Streamlines business processes and operations.
- Helps in managing finances, employees, customer relationships, and supply chain.
- Often integrated with other enterprise systems for seamless data flow.



# Conclusion

| Category                    | Description                                      | Examples                                     |
|-----------------------------|--------------------------------------------------|----------------------------------------------|
| <b>System Software</b>      | Manages hardware and system resources            | Operating Systems, Device Drivers, Utilities |
| <b>Application Software</b> | Performs specific user-driven tasks              | MS Office, Web Browsers, Video Games         |
| <b>Development Software</b> | Tools for creating and maintaining software      | IDEs, Compilers, Version Control Systems     |
| <b>Middleware</b>           | Connects different software systems              | Databases, Web Servers, Messaging Systems    |
| <b>Embedded Software</b>    | Software running on embedded systems             | Firmware, IoT Devices, Microcontrollers      |
| <b>Network Software</b>     | Facilitates network communication and management | TCP/IP, Firewalls, Network Monitoring Tools  |
| <b>Business Software</b>    | Manages business operations and processes        | ERP, CRM, Accounting Software                |

# Operating System (OS)

An **Operating System (OS)** is system software that **acts as an interface between the computer hardware and the user.** It manages hardware resources, runs applications, and provides **essential services for computer programs.**

# Utility Program

A **Utility Program** is a type of system software designed to help users **maintain, optimize, and manage** their computers. These programs perform specific tasks that **improve system performance, enhance security, and assist in file management.**

# 1 . Functions of Utility Programs :

Utility programs serve various functions, including:

## 1.1 System Performance Optimization

- Helps improve the speed and efficiency of the system.
- Includes tools like **Disk Cleanup, Disk Defragmenter, and Task Manager.**

## 1.2 File and Disk Management

- Organizes, compresses, and backs up files.
- Examples: **File Explorer, WinRAR, Disk Partitioning Tools.**

## 1.3 Security and Protection

- Protects against malware, viruses, and unauthorized access.
- Examples: **Antivirus software, Firewall, Encryption tools.**

# Utility Program

## 1.4 Data Backup and Recovery

- Helps users recover lost or deleted files.
- Examples: **Windows Backup, Acronis True Image.**

## 1.5 Networking and Communication

- Manages network connections and optimizes internet usage.
- Examples: **Wi-Fi Analyzers, Network Monitoring Tools.**

### 3. Importance of Utility Programs

- **Improve system performance.**
- **Enhance security and data protection.**
- **Assist in troubleshooting and problem resolution.**
- **Optimize disk and file storage.**



## 4. Built-in vs. Third-Party Utility Software

| Category           | Built-in Utility (Windows/Mac/Linux)           | Third-Party Utility        |
|--------------------|------------------------------------------------|----------------------------|
| Antivirus          | Windows Defender, Gatekeeper (Mac)             | McAfee, Norton             |
| File Compression   | Built-in ZIP Utility                           | WinRAR, 7-Zip              |
| Disk Cleanup       | Disk Cleanup (Windows)                         | CCleaner                   |
| Network Monitoring | Task Manager (Windows), Activity Monitor (Mac) | Wireshark                  |
| Backup & Recovery  | Windows Backup                                 | Acronis True Image, Recuva |

# Device Driver

## 1. What is a Device Driver?

A **Device Driver** is a specialized software that allows an operating system (OS) to communicate with hardware components. It acts as a bridge between the OS and hardware devices such as printers, keyboards, graphics cards, and network adapters.

### Example

When you connect a new printer to your computer, the OS installs a printer driver that enables communication between the OS and the printer.

## 2. Functions of a Device Driver

Device drivers perform several key functions, including:

- 1. Hardware Communication** – Acts as an intermediary between the OS and hardware.
- 2. Command Execution** – Converts OS instructions into hardware-specific commands.
- 3. Error Handling** – Detects and resolves hardware-related issues.
- 4. Resource Management** – Allocates system resources efficiently.
- 5. Performance Optimization** – Ensures the hardware runs efficiently.

# 3. Types of Device Drivers

## 3.1 Kernel Mode Drivers

- Operate at a low level, interacting directly with hardware.
- Provide high-speed communication but can crash the system if faulty.
- Examples:
  - Graphics Card Drivers
  - Network Card Drivers
  - Disk Controller Drivers

## 3.2 User Mode Drivers

Run in a separate space from the OS kernel.

- Less risk of crashing the system but may have slower performance.

Examples:

- Printer Drivers
- USB Device Drivers
- Audio Drivers

# 4. Common Device Driver Categories

| Category           | Examples                            |
|--------------------|-------------------------------------|
| Display Drivers    | NVIDIA, AMD, Intel Graphics Drivers |
| Printer Drivers    | HP, Canon, Epson Printer Drivers    |
| Network Drivers    | Ethernet, Wi-Fi, Bluetooth Drivers  |
| Storage Drivers    | Hard Disk, SSD, USB Drivers         |
| Audio Drivers      | Realtek, Sound Blaster Drivers      |
| Peripheral Drivers | Keyboard, Mouse, Scanner Drivers    |

## 5. Issues Caused by Faulty or Outdated Drivers

- **Hardware Malfunction** – Devices may not work properly.
- **System Crashes (Blue Screen of Death)** – Corrupt drivers can cause blue screen errors.
- **Performance Issues** – Old drivers may slow down performance.
- **Security Vulnerabilities** – Outdated drivers can be exploited by malware.

## 6. Future Trends in Device Drivers

1. **AI-Optimized Drivers** – Smart drivers that self-optimize based on system performance.
2. **Cloud-Based Drivers** – Remote driver management without manual installation.
3. **Universal Drivers** – Cross-platform drivers that work on multiple OS.



# Conclusion

Device drivers are crucial for seamless communication between the **OS** and **hardware**. They come in different types, including **kernel mode**, **user mode**, and **virtual drivers**. Regular updates and proper maintenance ensure system **stability, security, and optimal performance**.

Thank You  
Any Questions?