What Is an operating system

1. Main Purpose

- It acts as a bridge between users, applications, and the computer hardware.

- Without an OS, you would need to control hardware manually, the OS makes that unnecessary by handling all the low-level details for you.

2. Key Responsibilities

- Process Management: Decides which programs run, for how long, and in what order.

-Memory Management: Keeps track of what part of memory is used by which program and ensures efficient usage.

-File System Management: Organizes data into files and directories so you can store, find, and retrieve information easily.

-Device Management: Controls input/output devices (keyboard, mouse, printer, etc.).

-Security & Access Control: Protects data and resources from unauthorized access.

-User Interface: Provides a way for you to interact with the computer (graphical interface like Windows, or text-based like a terminal).

3. Examples

-Desktop/Server OS: Windows, macOS, Linux (Ubuntu, Fedora, etc.)

-Mobile OS: Android, iOS

-Embedded OS: FreeRTOS, VxWorks (for IoT and devices)

What Is Hardware Compatibility

1. Source & Licensing

- Windows: Proprietary, closed-source (owned by Microsoft). Requires purchasing a license.

- macOS: Proprietary, closed-source (owned by Apple). Comes pre-installed on Apple hardware.

- Linux: Open-source and free to use. Anyone can view, modify, and distribute the code.

2. Hardware Compatibility

- Windows: Works on a wide range of hardware from many manufacturers (Dell, HP, Lenovo, custom PCs).

- macOS: Officially runs only on Apple hardware (MacBooks, iMacs).

- Linux: Runs on nearly any hardware, from old PCs to servers to Raspberry Pi boards.

3. User Interface (UI)

- Windows: Uses the Start Menu, Taskbar, and Windowed interface. Familiar to most users.

- macOS: Uses the Dock and Menu Bar with a very consistent, polished interface focused on simplicity.

- Linux: UI depends on the desktop environment (GNOME, KDE, XFCE, etc.). Highly customizable.

4. Software & App Availability

- Windows: Best support for commercial software and games.

- macOS: Strong support for creative software (Final Cut Pro, Logic Pro) but fewer games than Windows.

- Linux: Massive library of free/open-source software, but some commercial apps (like Adobe Suite) may not have native versions.

5. Security & Malware

- Windows: Most targeted by malware due to its popularity, so antivirus software is recommended.

- macOS: Fewer viruses, but still vulnerable; relies on Apple’s built-in protections (Gatekeeper, XProtect).

- Linux: Considered very secure, partly because of its permission system and smaller desktop market share (less targeted).

Why are there so many Linux distros?

Linux is open source, meaning anyone can modify the source code and create their own version.

Different groups or companies focus on different needs: user-friendliness, performance, security, server use, privacy, or lightweight systems for old computers.

This leads to hundreds of distros, each with its own package manager, desktop environment, and philosophy — but they all share the Linux kernel.

1. Debian-based

-Focus: Stability, reliability, large software repository.

Popular Examples:

- Debian – The parent project, known for stability.

- Ubuntu – Most popular desktop distro, very user-friendly.

- Linux Mint – Beginner-friendly, based on Ubuntu but with a Windows-like interface.

3. Arch-based

Focus: Rolling releases, bleeding-edge software, DIY approach.

Popular Examples:

- Arch Linux – Highly customizable, for advanced users who want full control.

- Manjaro – User-friendly Arch-based distro, easier installation and updates.

- EndeavourOS – Lightweight Arch-based distro with minimal preinstalled software.

Vocab Section

HID (Human Interface Device)

A device that lets humans interact with a computer. It’s usually a standard USB device type.

Examples: Keyboard, mouse, game controller.

Serial

A communication method where data is sent one bit at a time over a single channel.

Examples: Arduino’s Serial Monitor, UART communication.

SPI (Serial Peripheral Interface)

A fast, synchronous communication protocol is used to connect microcontrollers to peripherals.

Examples: Connecting an SD card to an Arduino, reading data from a temperature sensor, interfacing with an OLED display.

Analog Signal

A continuous signal that can take on any value in a range.

Examples: Audio waveforms, temperature sensor output voltage, dimming of a light bulb with a potentiometer.

Digital Signal

A signal that has only two states: ON/OFF or HIGH/LOW.

Examples: Computer binary data, LED on/off control, clock pulses in a microcontroller.

Nyquist Rate

The minimum rate at which a signal must be sampled to avoid losing information (must be at least twice the highest frequency of the signal).

Examples: Audio sampled at 44.1 kHz (captures frequencies up to ~22 kHz), radar signal processing, image scanning resolution.

Hardware

The physical components of a computer or electronic system.

Examples: CPU, RAM stick, Raspberry Pi board.

Software

Programs and code that tell hardware what to do.

Examples: Microsoft Word, Google Chrome, MATLAB.

Firmware

Software stored permanently in hardware (often in ROM/flash memory) that controls low-level functions.

Examples: BIOS on a PC, firmware in a router, microcontroller bootloader.

Bash Scripting

Writing scripts in the Bash shell language to automate tasks on Linux/Unix systems.

Examples: Automating file backups, running a list of commands with one script, writing a build script for code compilation.

.bin File

A binary file that contains raw data, often firmware, executables, or other compiled code meant to be read by machines, not humans.

Examples: Router firmware update file, game ROMs, raw disk images.

Ghidra Software Suite

An open-source reverse engineering tool developed by the NSA, used to analyze binaries and understand how programs work.

Examples: Disassembling a .bin firmware file, reverse engineering malware, analyzing compiled code for security research.

Binwalk

A tool used to analyze, extract, and reverse engineer firmware images or binary files.

Examples: Extracting files from a router firmware, finding embedded file systems in IoT devices, inspecting raw binary data.

Jadx

A decompiler that converts Android APK files into human-readable Java source code.

Examples: Viewing source code of an Android app, analyzing open-source apps, checking for malicious code.

dex2jar

A tool that converts Android’s .dex (Dalvik Executable) files into Java .class files so they can be opened with a Java decompiler.

Examples: Converting an APK’s classes.dex file, preparing Android code for analysis in JD-GUI, debugging custom Android builds.

JD-GUI

A graphical tool (Java Decompiler) used to view Java source code from compiled .class files or .jar archives.

Examples: Inspecting Java libraries, viewing source of converted dex2jar output, learning how a program works.

APK (Android Package)

The file format used to distribute and install applications on Android devices.

Examples: WhatsApp.apk, YouTube.apk, custom Android game installs.