What is Operating System Architecture

Dina Sharifi

If we consider the hardware to be the skeleton of our computers, we would need to understand what makes them useful, operational. Operating System architecture can explain how computers run software and manage resources, how various parts of the OS make the hardware usable, and applications—functional. Most operating systems share common software structures: the kernel, process management, memory management, file systems, and input/output control.

The core of every OS is the kernel. It manages the communication between hardware and software. For instance, when we press a key, the kernel ensures that the signal is understood and passed to the correct program by the CPU. OSs can be for desktops (e.g. Windows, Linux, macOS), and for mobile (e.g. iOS, Android). Desktop OS have kernels that can handle heavy workloads and have no trouble multitasking, whereas mobile systems' kernels are optimised for low power consumption and speed.

Every application that is opened is a process, and the OS chooses how much CPU time to devote to each. Without the process management, one process could dominate the CPU and freeze the system. Computers use this to manage a number of programs at once, while phones use it to keep apps responsive while running in the background (e.g. music or notifications).

Memory management means efficient use of the RAM. The OS tracks the parts of memory in use, and allocates space to different processes, moving data between RAM and storage if needed. This allows smooth multitasking among programs. In smartphones, it is more crucial, since the RAM is smaller, and the OS usually suspends some apps to save space and battery.

The file system is the way in which the OS organises and saves data on permanent storage. It is responsible for how files are named, accessed, and secured. Both in computers and smartphones, the file system tracks the data, applications, and system files, to make sure they don't interfere with each other.

The input/output management allows the OS to coordinate with external devices. For example, on computers, this can refer to handling keyboards, USB devices, etc. On smartphones, it can manage the sensors, GPS, cameras.

There's also a user interface in operating systems. It's the layer that enables humans to interact with the system. On computers, it can be the graphical interface with windows and icons, or a command-line interface for direct interaction with the OS. Smartphones have touch-based UI's with on-screen icons and keyboards, designed for smaller screens.

And, of course, there's security and resource management, with which the OS protects memory. This enforces permissions to keep files safe, and manages networking securely. On phones, this can include sandboxing — isolating the applications in a restricted environment, to limit their access to system resources and user data — which protects sensitive information.

The evolution of operating systems has always occured alongside hardware. The systems have transformed from being simple control programs, into today's highly complex systems, capable of multitasking, networking, and having advanced user interfaces. Due to the rise of mobile devices, new goals were established - efficiency, touch input, connectivity - which reshaped OS architecture into what we daily on our devices.

In short, OS architecture shows how software makes hardware usable. Whether one uses Windows, Linux, Android or iOS, the same principles apply to all: allocating processes, memory, files, devices, and user security. And similar to hardware, the only difference lies in how each system balances power and performance.