

Varonkas notes on OS Architecture.

Think of your laptop booting up. You press the power button and the screen lights up, the keyboard responds and a login screen appears. That moment is your operating system taking control. It decides what runs, how memory is used and who talks to what.

Every thing that happens between your hardware and your apps passes through it.

For that we need the kernel / the core / of the OS. The kernel takes signals from programs like "open this file" or "connect to WiFi", and translates them into actions the hardware understands. It sends instructions to the CPU, moves data in and out of memory, and uses drivers to talk to devices. It's the traffic controller, the godfather. Nothing happens without it.

Now, different situations need different kinds of control. A smart-watch doesn't need the same kernel a server needs. It needs a smaller, specialized kernel, that doesn't waste power. A server, on the other hand, runs hundreds of background things. We need a heavy kernel for that, with advanced scheduling, memory protection, security layers and everything.

The kernel handles several key systems.

Process management: Deciding what goes to CPU right now. For example, when you stream music while typing, the OS rapidly switches between both tasks so it feels simultaneous.

Memory Management: Keeping programs from colliding. When your browser eats too much RAM, the OS moves unused data to disk. For it to not crash,

Device management: letting hardware and software communicate. You plug in a game-pad and the OS loads a driver so the kernel knows what to command the hardware.

There are also different kernel architectures. A monolithic kernel (like linux) keeps most of these systems in one big piece of code. This is fast but hard to debug. A microkernel (Mikix or QNX) only handles bare minimum in the core and runs other services separately. Slower But easier to maintain. So yeah, OS architecture is figuring out what must happen now, what can wait and how everything fits together so the processes stay stable.