A context switch saves the full CPU state of the old thread and loads the saved state of the new one. This includes registers, the program counter, the stack pointer, and flags. The OS may also update the TLB if the next thread uses another address space.

When many threads wake at once, the scheduler checks its run queues and picks threads by its policy. It tries to give fairness while also keeping high-priority threads first. Burst wakeups can change queue order and force quick decisions.

A page fault happens when a virtual address has no valid mapping. The CPU traps into the kernel. The OS looks up the page table entry and loads the page from disk or memory, fixes the entry, and resumes the thread.

A lock can cause spikes when many cores touch the same cache line. The line bounces between cores and slows down the path. Even small critical sections can stall threads if they try to enter at the same time.

A system call crosses the user boundary with a trap instruction. The CPU switches to kernel mode, the OS checks the call, runs it with full privilege, then returns to user mode. This protects the system.