

RTOS Scheduling

Electrical Engineering

What a real-time kernel provides, and when a superloop is still enough.

An RTOS shares the CPU between tasks and protects the deadline of the most urgent one.

CONTEXT SWITCH

A few microseconds

TICK RATE

Often 1 kHz

STACK PER TASK

Sized, then measured

- Each task has its own stack and priority
- Preemptive: the highest ready task runs
- Blocking on a queue or semaphore yields
- The idle task is where sleeping belongs
- Priority inversion needs inheritance to fix
- Determinism matters more than throughput

LOOK FOR IT

A superloop copes until two activities need different rates and neither may block the other.

An RTOS does not make code faster, only its timing predictable. If a superloop already meets every deadline, it is enough.