A run of the scheduler consists of pushing the context of task $i$ when it comes in but popping the context of task $j$ on its way out.
L5: Scheduling Algorithms

1. Asynchronous Interrupt Event Driven
   • One “dummy” forever loop.
   • A few Interrupt Service Routines (ISR) – Interrupt handlers.
   • Simplest case of Background / Foreground scheduling – no background tasks.

2. Polled & Polled with Timing Element
   • One continuous loop testing the input signal – looks for changes.
   • Fastest scheduling algorithm.
   • Deterministic (Computable and bounded) – if only one event is handled.
   • Worst case? Time to loop.
   • Timing element just adds a delay after the event is true to make event period constant.
   • Case of F/B – no foreground.
3. **State Based**
   - Scheduler is a state machine.
   - Implemented with “case” statements or if-then-else’s.
   - Drawback: complexity grows with # of inputs.

4. **Synchronous Interrupt Event Driven**
   - Basis for time-shared systems
   - An internally generated timing signal causes an interrupt.
   - An ISR switches to another task.
   - Signal can be periodic (fixed rate) or aperiodic (sporadic).
L5: Scheduling Algorithms

5. Combined Interrupt Event Driven
   • Combines synchronous and asynchronous interrupts.

6. Foreground - Background
   • Background tasks have low priority and can be interrupted at any time.
   • Foreground tasks are interrupt-driven and implement the RT aspect of the system (if any).
7. Time-Shared
   • Tasks may or may not be equally important.
   • Tasks with same time slice → periodic scheduler.

a) First-Come First-Served (FIFO) (RTAI ✓)

   - Non-preemptive
   - RT issues
b) Shortest Job First
   – CPU usage is placed in TCB and used to determine order of TCB’s.
   – Estimates of how much time the job will need when next given the CPU.
   – Preemptive or not. If preemptive, then a new shorter task can interrupt the current one.

c) Round Robin (RTAI ✓)
   – FIFO with preemption
   – Time Quantum = Time Slice
   – CPU allocated to a process for a time slice. If done, CPU is released. If not (during t. slice), process interrupted when time expires and put at the end of queue.
L5: Scheduling Algorithms

c) Round Robin
Assignment

• Read Sections 12.3 – 12.4