Rate Monotonic Scheduling
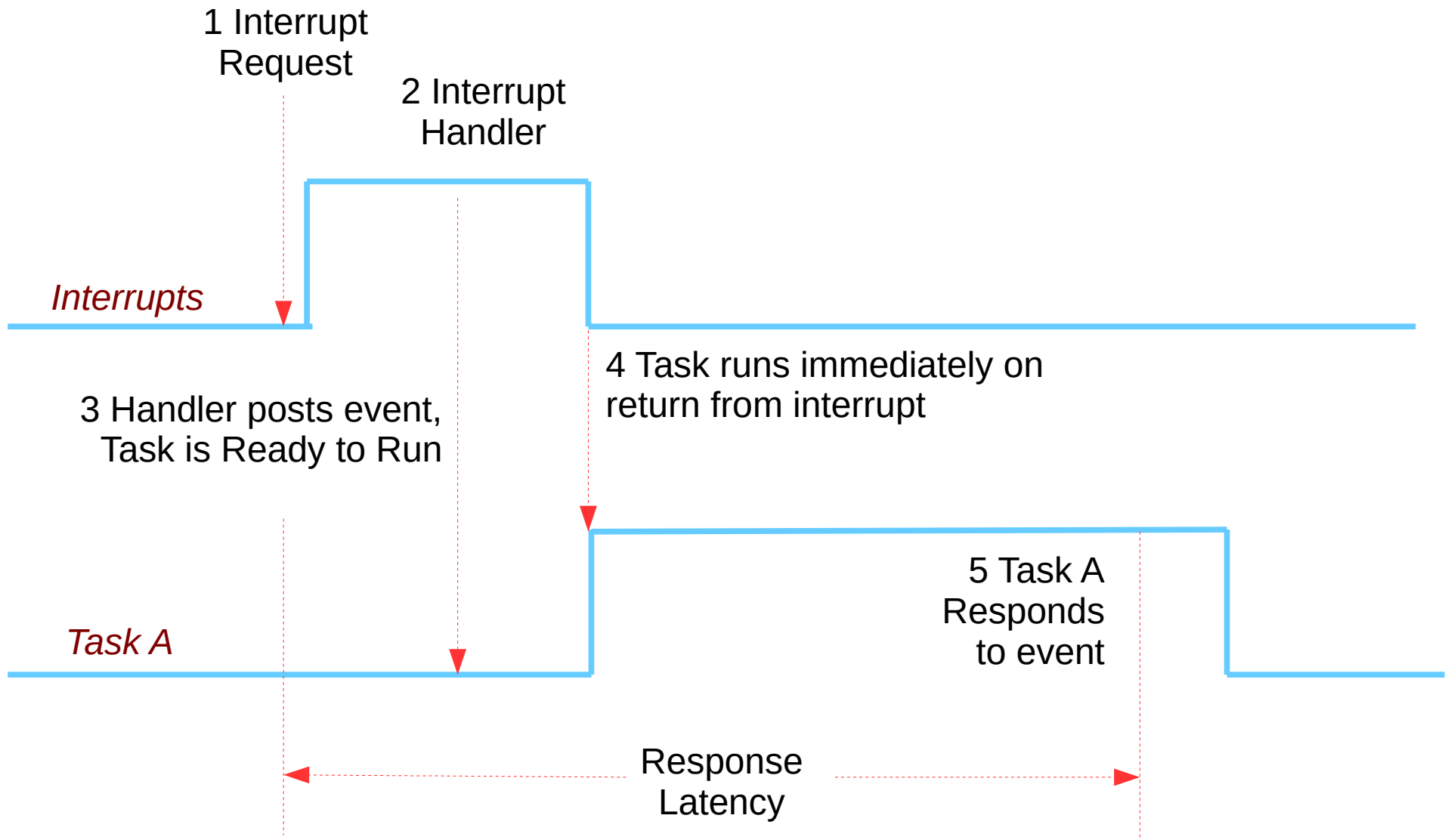
Assumption:

- *No resource sharing (processes do not share resources, e.g. a hardware resource, a queue, or any kind of semaphore blocking or non-blocking (busy-waits))* Wikipedia "Rate Monotonic Scheduling"
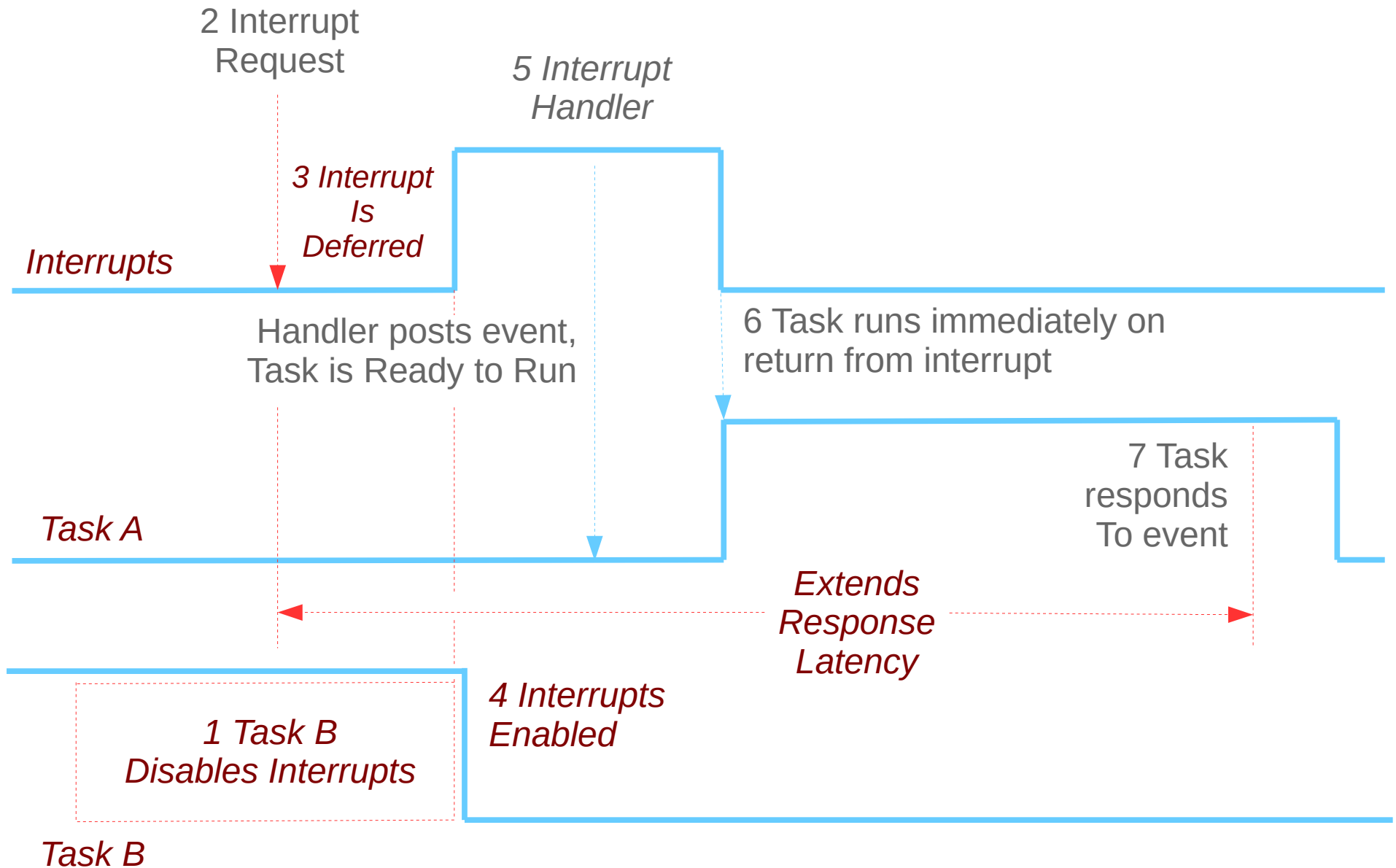
Real world: We must protect shared resources with locks of some kind. The most aggressive: (1) Disabling interrupts, and (2) Disabling pre-emption.

What are the effects of real-time performance when this assumptions is violated?

Normal Interrupt Processing

1 Interrupt Request

2 Interrupt Handler

Interrupts

3 Handler posts event, Task is Ready to Run

4 Task runs immediately on return from interrupt

Task A

5 Task A Responds to event

Response Latency

Effect of Disabling Interrupts

2 Interrupt
Request

5 Interrupt
Handler

3 Interrupt
Is
Deferred

Interrupts

Handler posts event,
Task is Ready to Run

6 Task runs immediately on
return from interrupt

7 Task
responds
To event

Task A

Extends
Response
Latency

1 Task B
Disables Interrupts

4 Interrupts
Enabled

Task B

Effect of Disabling Pre-emption

2 Interrupt
Request

3 Interrupt
Handler

5 Interrupt returns to
interrupted task
with no switch

Interrupts

4 Handler posts event,
*Task is cannot run.
Placed in pending state*

7 Pending task cannot
Run until Pre-emption
Re-enabled

8 Task
responds
To event

Task A

*Extends
Response
Latency*

1 Task B Disables
Pre-Emption

6 Task B Enables
Pre-Emption

Task B