

Definition 1 (Hard real-time)

Absolutely no deadline can be missed (it could have health, financial or ecological consequences).

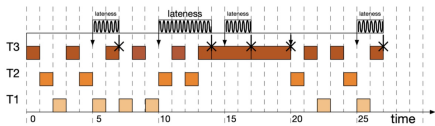
Definition 2 (Firm real-time)

Ideally no deadline should be missed. However, some deadline misses may be tolerated, in which case we define a Quality of Service (QoS). For example, in video conferencing or live streaming, the system must ensure that video frames are transmitted and displayed within a specific time frame to maintain a smooth and uninterrupted video stream. Any delay in transmitting or displaying video frames can result in jitter, buffering, or even loss of data.

Definition 3 (Soft real-time)

Deadlines can be missed.

**Round-Robin [24].** The Round Robin illustrated in Figure 2.2 is a technique where each task periodically receives an equal share of the processing resource — e.g. the CPU — in order to provide a good average process response time. The key idea is to assign time slices in equal portions to each process in a circular order. This technique is a typical scheduling scheme for most general purpose operating systems (unix, linux, windows). Figure 2.2 shows that in our set of tasks, the Round Robin is deadly for  $\tau_3$ .



Task	Execution time	Period (& deadline)	Load
$\tau_1$	4	20	20%
$\tau_2$	2	10	20%
$\tau_3$	3	5	60%
Total			100%

**Rate Monotonic [18].** Rate Monotonic is a preemptive scheduling algorithm used in Real-Time Operating System (RTOS), where each task  $\tau$  receives a static priority  $P_\tau$ . This priority is assigned to the task by looking at its periodicity: the shorter the period, the higher the priority. In our example system,  $\tau_3$  receives the highest priority,  $\tau_2$  the middle priority and  $\tau_1$  the lowest priority. The schedule produced is shown in Figure 2.3. In our example, we observe that with the RM scheduler, all tasks are processed on time. Note that the schedule repeats from time marker 20 (the least common multiple of the task's periods).

