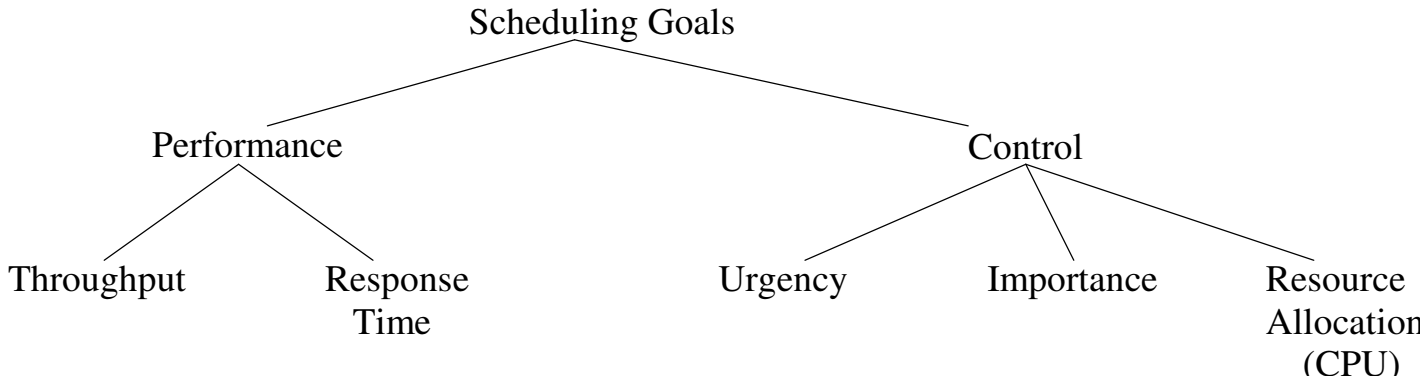

1) Liu and Layland ('73) theorems to analyze a periodic hard-real-time system under fixed-priority scheduling:

- If the threads will meet their deadlines under any fixed priority assignment, then they will do so under an assignment that prioritizes threads with shorter periods over those with longer periods. This policy is known as *rate-monotonic scheduling*.
- To check that deadlines are met, it suffices to consider the worst-case situation, which is that all the threads' periods start at the same moment.

Consider three periodic threads which do not interact, but have the following characteristics:

Thread	Period and Deadline	Worst-case execution
T1	7	2
T2	5	1
T3	6	3

a) Draw the Gantt chart to determine if the rate-monotonic schedule will work.

2) A dynamic-priority scheduling scheme for hard-real-time system is Earliest Deadline First (EDF) scheduling. In EDF, each time a thread becomes runnable, you reassign priorities such that the sooner a thread's next deadline, the higher its priority. Consider two thread example that did not work with fixed priority scheduling:

Thread	Period and Deadline	Worst-case execution
T1	4	2
T2	6	3

a) Draw the Gantt chart to determine if the EDF schedule will work.

3) An examples where we might want to dynamically adjust priorities: have the OS adjust disk-bound threads (e.g., virus scan) toward higher priorities and CPU/processor-bound threads (e.g., graphics rendering) toward lower priorities.

a) Which performance criteria does this improve: throughput or response time?

b) Explain why performance is improved.

4) An examples where we might want to dynamically adjust priorities: have the OS adjust interactive threads (e.g., word processor) toward higher priorities and CPU/processor-bound threads (e.g., graphics rendering) toward lower priorities.

a) Which performance criteria does this improve: throughput or response time?

b) Explain why performance is improved.

5) How can we generalize the two previous examples?