DM timeline with preemption:
IMG11-1

Dynamic priority:

- We use EDF(Earliest Deadline First)
    - Assign priority according to current absolute deadline
        - Based on current task mix and the current time
        - Time is no longer an offset
    - Answer the question: whose deadline is closest to now?
    - If we have less than 100% utilization(of our processor), this is optimal scheduling
        - It produces the best possible real-time schedule
    - Very CPU intensive

    Implementation strategies:

    1) LST (Least Slack Time) (less slack time, higher priority)
        - Assign priority based on d-t-x

            d = deadline
            t = curr time
            x = execution time left
        - Assign priority at task injection (non-strict LST)

            - or -
        - Use a timer to trigger priority adjustment
            - As time goes by our slack time decreases
        - If non-preemptive, EDF ≡ LST
            - Because execution time left never changes
    2) Execution time left is hard to measure
        - Use deadline & injection time to determine priority
        - Order tasks based on time to deadline
        - Task’s priority “automatically” goes up as time goes by

        Implementation:

        - We scan our Queue at injection time for location based on time to deadline
        - As we scan we can update existing times
            - We store injection time & deadline with each task
                - Adjust priority based on these 2 values