EDF

# Earliest Deadline First Scheduling

Dynamically assigning priorities to threads

When threads become runnable; re-assign priorities according to rule:

The sooner the deadline, the higher the priority

Optimal scheduling algorithm

If a collection of independent jobs (with arrival time, execution requirement and deadline), can be scheduled so they all reach their deadlines. EDF will schedule them so they all reach their deadline.

When we encounter a tie in priority you either

1. Break priority in favor of the already running thread
   1. Correct way in practice
2. Use any other tie-breaking strategy
   1. Theoretical

Utilization

Ex.

|  |  |  |
| --- | --- | --- |
| Thread | Execution time | Deadline |
| T1 | 1 | 8 |
| T2 | 2 | 5 |
| T3 | 4 | 10 |

If it’s under 100% then it is schedulable with EDF

Running example with EDF scheduling:

At time 0 T2 is prioritized over T1 and T3 (5 versus 8 and 10), and it finishes at time 1. Now, we run T2 (8 versus 10 and 10) that finishes at 3. Here we have a tie in priorities, because both T2 and T3 has a deadline at 10. We can use any tie-breaking strategy, so we choose the Thread with the shortest execution time (T2). T2 finishes at time 5. Now we run T3 (10 versus 15 and 16) which finishes at 9. All threads reached their deadlines and would keep on doing so if we kept on going.